

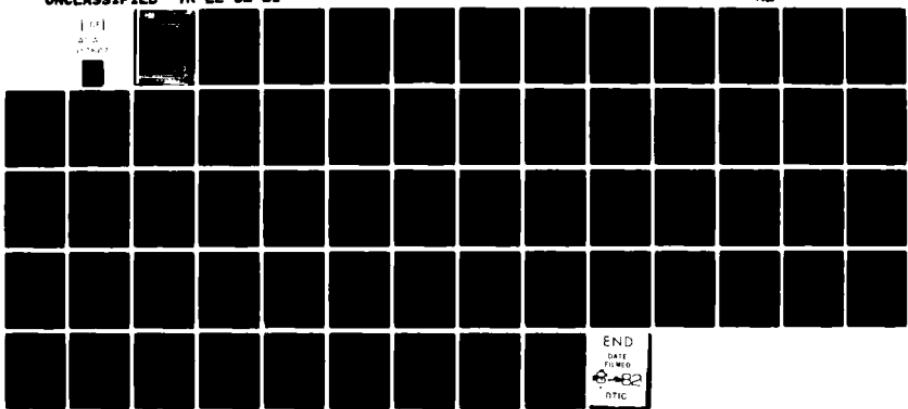
AD-A117 567

PURDUE UNIV LAFAYETTE IN SCHOOL OF ELECTRICAL ENGINEERING F/0 9/5
FAULT DIAGNOSIS OF NONLINEAR ANALOG CIRCUITS. VOLUME I: DC DIAG—ETC(U)
JUL 82 P M LIN, Y S ELCHERIF N00014-81-K-0323

UNCLASSIFIED TR-EE-82-21

ML

100
AIA
1982



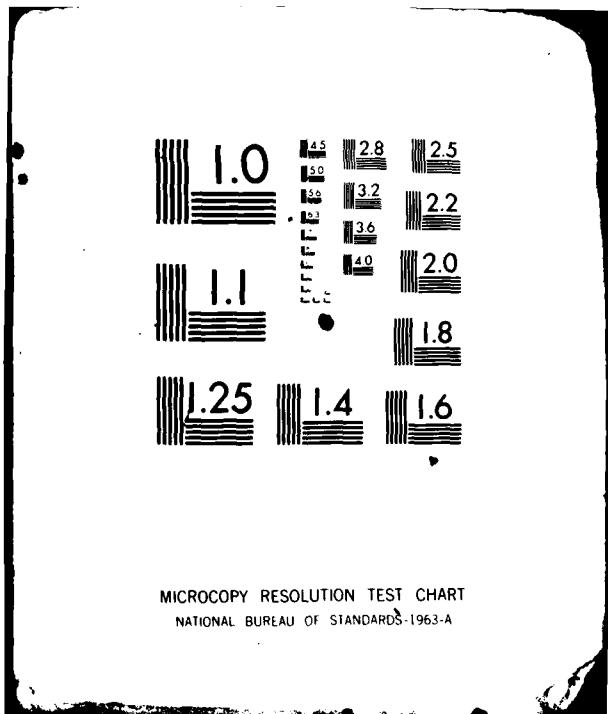
END

DATE

FILED

8-482

RTIC



AD A117567

FAULT DIAGNOSIS OF NONLINEAR ANALOG CIRCUITS

VOLUME I

DC DIAGNOSIS OF HARD FAILURES



P. M. Lin
Y.S. Elcherif

TR-EE-82-21
July 1982

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Refill and/or	<input type="checkbox"/>
Dist. Special	<input type="checkbox"/>
A	

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

This work was supported by Office of Naval Research,
Contract No. N00014-81-K-0323.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-EE 82-21	2. GOVT ACCESSION NO. AD-A017567	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Fault Diagnosis of Nonlinear Analog Circuits. Volume 1, DC Diagnosis of Hard Failures		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) P. M. Lin, Y. S. Elcherif		8. CONTRACT OR GRANT NUMBER(s) N00014-81-K-0323
9. PERFORMING ORGANIZATION NAME AND ADDRESS Purdue University School of Electrical Engineering W. Lafayette, Indiana 47907		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia		12. REPORT DATE July 1982
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		13. NUMBER OF PAGES 58
		15. SECURITY CLASS. (of this report) Unclassified
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release. Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) SAME		
18. SUPPLEMENTARY NOTES None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Fault Diagnosis Fault Dictionary DC Tests		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The dc fault dictionary approach has been found very useful for detecting hard failures (open circuits and short circuits) in nonlinear analog circuits. In this research project, we have introduced several new techniques to enhance the capability of the dc fault dictionary. Among our new results, the most significant ones are: (1) The use of the complementary pivot method to find test node voltages under all faults. (2) The set-up of the fault dictionary in the form of integer codes instead of test node voltages. The former technique makes it possible to rapidly analyze large circuits under all fault conditions.		

The latter technique eliminates all arithmetic operations after test. Thus, after the test on a faulty circuit has been made, the technician need only look up literally speaking, the fault dictionary to determine the fault.

This technical report contains a detailed description of the background material and the new algorithms. The algorithms have been implemented in a digital computer program. The result of the diagnosis of a video amplifier circuit by the computer program is included in the report as an example.

A companion report, TR-EE 82-22, describes a new multifrequency method for soft failure analysis.

TABLE OF CONTENTS

	<i>page</i>
Abstract	1
1. Introduction	2
2. Review of an Existing DC Fault Dictionary Approach	4
2.1 Pretest Analysis	4
2.2 Ambiguity Sets, Fault Isolation, and Fault Dictionary	7
2.3 Post-test Analysis	10
3. Proposed New Approaches	11
4. Modeling of Piecewise Linear Resistive Networks with Ideal Diodes	13
4.1 Piecewise Linear Approximation to a Smooth Curve	13
4.2 A Simple PWL Resistive Network and Its Solution	15
5. The Complementary Problem and Methods of Solution	20
5.1 The Complementary Problem	20
5.2 The Exhaustive Search Method	21
5.3 Lemke's Complementary Pivot Method	24
6. Prestes DC Analysis Using Complementary Pivot Method	30
6.1 The Use of Switches	30
6.2 Formulation of Multiport Constraint Equations	30
6.3 Calculation of Test Node Voltages	33
6.4 A Simple Example	35
7. Fault Isolation and Reduction of Test Nodes	39
7.1 Construction of the Table of Ambiguity Sets	39
7.2 Fault Isolation by Intersection of Ambiguity Sets	42
7.3 Reduction of the Number of Test Nodes	44
8. An Integer-Code Fault Dictionary and Post-test Fault Identification	47
8.1 Compilation of the Fault Dictionary	47
8.2 Post-test Fault Identification by Dictionary Look Up	50
9. A Video Amplifier Circuit Example	52
10. Conclusions and Further Research	56
References	57

ABSTRACT

The dc fault dictionary approach has been found very useful for locating hard failures (open circuits and short circuits) in nonlinear analog circuits. In this research project, we have introduced several new techniques to enhance the capability of the dc fault dictionary. Among our new results, the most significant ones are: (1) The use of the complementary pivot method to find test node voltages under all faults. (2) The set-up of the fault dictionary in the form of integer codes instead of test node voltages. The former technique makes it possible to rapidly analyze large circuits under all fault conditions. The latter technique eliminates all arithmetic operations after test. Thus, after the test on a faulty circuit has been made, the technician need only look up, literally speaking, the fault dictionary to determine the fault.

This technical report contains a detailed description of the background material and the new algorithms. The algorithms have been implemented in a digital computer program. The result of the diagnosis of a video amplifier circuit by the computer program is included in the report as an example. A companion report, TR-EE-82-22, describes a new multifrequency method for soft failure analysis.

1. INTRODUCTION

This research report consists of two volumes. The present volume describes new approaches to dc test for identifying hard failures. The second volume describes ac multifrequency test for identifying soft failures. Our philosophy in fault diagnosis is to go through dc test first, and then followed by ac test when necessary. Since hard failures account for about 80% of the faults in analog equipment, the dc test, which is easier to perform than ac test, will be adequate for the majority of the cases. For the remaining more difficult cases of soft failures, we rely on the ac multifrequency test method.

In dc fault diagnosis of analog circuits, one applies dc inputs and measures dc outputs (voltages and/or currents). From the measured results one attempts to locate the faulty components. Two approaches have been described in the literature for achieving this goal:

(1) The component simulation approach [1]. This method is based on the dc small signal equivalent circuit of the given nonlinear circuits. The circuit model used in the analysis is linear and resistive, since capacitances behave as open circuits and inductances as short circuits in dc steady state. A set of small signal network functions (short circuit admittances y_{ij} are considered in [1]) are expressed in terms of element values some or all of which are unknowns. The measured values and the symbolic expressions for these network functions together lead to a set of nonlinear algebraic equations. The unknown element values (r_j , g_m , etc.) are then found by the use of Newton iteration or some minimization technique [1]. The faulty components are then identified by determining whether the circuit's elements values are within or outside the design tolerance margin.

It is clear that in this approach a large amount of computation must be done *after* the testing is made. For this reason, it is also called a simulation-after-test approach.

As described in [1], this approach assumes the availability of various network functions in symbolic forms. This seriously limits its usefulness. At present, all known symbolic network analysis programs can only handle small networks in the order of 10 nodes and 35 branches [2]. Until a breakthrough in the area of symbolic network analysis happens, the approach of [1] is likely to remain of academic interest.

Instead of using the linear equivalent circuits, one can also consider the use of nonlinear circuit models in the simulation-after-test approach. Consider a diode for example. Instead of representing it by a linear incremental resistance r , as done in [1], we can also characterize it by the nonlinear model $I_D = I_S(e^{V_D} - 1)$, and solve for the unknown parameters (I_S, θ). The diagnosability problem of this approach has been investigated in [3], but a diagnosis algorithm is lacking at this time.

(2) The fault simulation approach [4]. This method is based on the *nonlinear*

circuit model. Various component values are changed so as to simulate component failures. The voltages of some judiciously selected test nodes are calculated under all possible component failures. These *calculated* voltages, after suitable organization (see Section 2 for details), constitute a "fault dictionary". After a test on the faulty circuit has been made, the *measured* voltages are compared with those stored in the fault dictionary. The fault is identified through the application of some fault location algorithm (e.g. minimum sum-of-squared-errors, as in [4]).

We see that in this approach practically all computations (in the form of dc analysis of a nonlinear circuit) are done before the testing of the faulty circuit. For this reason it is also called a simulation-before-test approach, or a "fault dictionary" approach. It should be pointed out that the fault dictionary concept need not be restricted to dc inputs and outputs. At least for *linear* circuits we could consider the use of ac inputs and outputs [5].

One obvious limitation of the fault dictionary approach, whether dc or ac, is its inability to treat all possible failures of a component. Consider for example a resistor R having 1000 ohms nominal value and 10% tolerance. Then when the value of R is changed, due to whatever reason, to within $(0, 950)$ or $(1050, \infty)$ ohms, the resistor is considered as faulty. Clearly it is impossible to repeat the analysis of the circuit under all possible failures even for this resistor alone. Therefore, in the fault dictionary approach we consider only catastrophic component failures (or hard failures), namely open circuits and short circuits. Thus the fault dictionary approach cannot diagnose faults other than open circuits and short circuits. Despite this limitation, the fault dictionary approach is useful because statistics shows that about 80% of failure in analog equipment are in the form of short and open circuits [4]. For the diagnosis of soft failures, we rely on the multifrequency test methods [6, 19].

The use of dc, instead ac inputs/outputs, brings forth some further limitations: short circuits of inductors and open circuits of capacitors are not detectable with the dc approach. Nevertheless, the dc method is preferred over the ac method in fault dictionaries for the following practical reasons: (a) dc measurements are easier to execute than ac measurements; (b) A dc analysis, is less expensive than an ac analysis (cost ratio approximately 1-2, see [7]); (c) The dc approach yields more *direct* information about the operating points of semiconductor devices.

The dc fault dictionary approach has been successful for circuits of moderate complexity, e.g., the video amplifier of [1]. When larger circuits are considered, two serious difficulties arise:

- (1) Both the storage and the computation time for the fault dictionary become prohibitive;
- (2) As the fault dictionary becomes less comprehensive, there is a greater chance of encountering a fault not included in the dictionary, and hence not identifiable.

The objective of the present research task is to develop new techniques to enhance the capability of the dc fault dictionary approach. In Section 2, we first give a more detailed description of an existing fault dictionary approach in order to see where the difficulties lie. In Section 3, we present an overview of our proposed new techniques. Sections 4 to 6 give the background material for the new approaches. Sections 7 and 8 describe new algorithms for compiling the fault dictionary. Section 9 gives a complete example of applying our new approach to a video amplifier. The documentation of the software is given in a separate technical report [20].

2. REVIEW OF AN EXISTING FAULT DICTIONARY APPROACH

In order to appreciate the significance of our proposed new approaches, it is necessary first to review what is currently being done in dc fault diagnosis. We shall use the video amplifier example of [4] as a vehicle to explain all phases of the dc fault dictionary approach. We then (in Section 3) point out where the difficulties lie and propose new techniques for overcoming these difficulties.

The dc fault dictionary approach consists of two distinct stages:

Stage 1. Pretest analysis to compile the fault dictionary.

In this stage the analog circuit is simulated by a digital computer program under nominal as well as all preselected catastrophic faults. Judiciously chosen dc input voltages are applied. The induced dc voltages at a selected set of test nodes are calculated. These voltages are then stored in the ATE (automatic test equipment) and constitute the fault dictionary.

Stage 2. Post-test analysis to identify the fault.

In this stage, measurements of test node voltages have been made on the circuit, and the measured values are compared with those stored in the fault dictionary. First, a fault detection algorithm is applied to determine whether the circuit is faulty at all. If the answer is affirmative then the fault is identified by the application of some fault isolation algorithm (e.g., minimum sum-of-squared errors, as in [4]).

2.1 Pretest Analysis

As an example, consider the video amplifier circuit of [4] whose schematic diagram is reproduced here as Figure 2.1. The amplifier has 9 transistors (Q1 through Q9), 7 diodes (D1 through D7), 4 zener diodes (DZ1 through DZ4), 45 resistors (R1 through R45), and 4 inductors (L1 through L4). The amplifier is constructed with discrete components. There are 43 nodes (excluding the common ground). Not all of these nodes are accessible. Otherwise the fault diagnosis problem would be trivial [8].

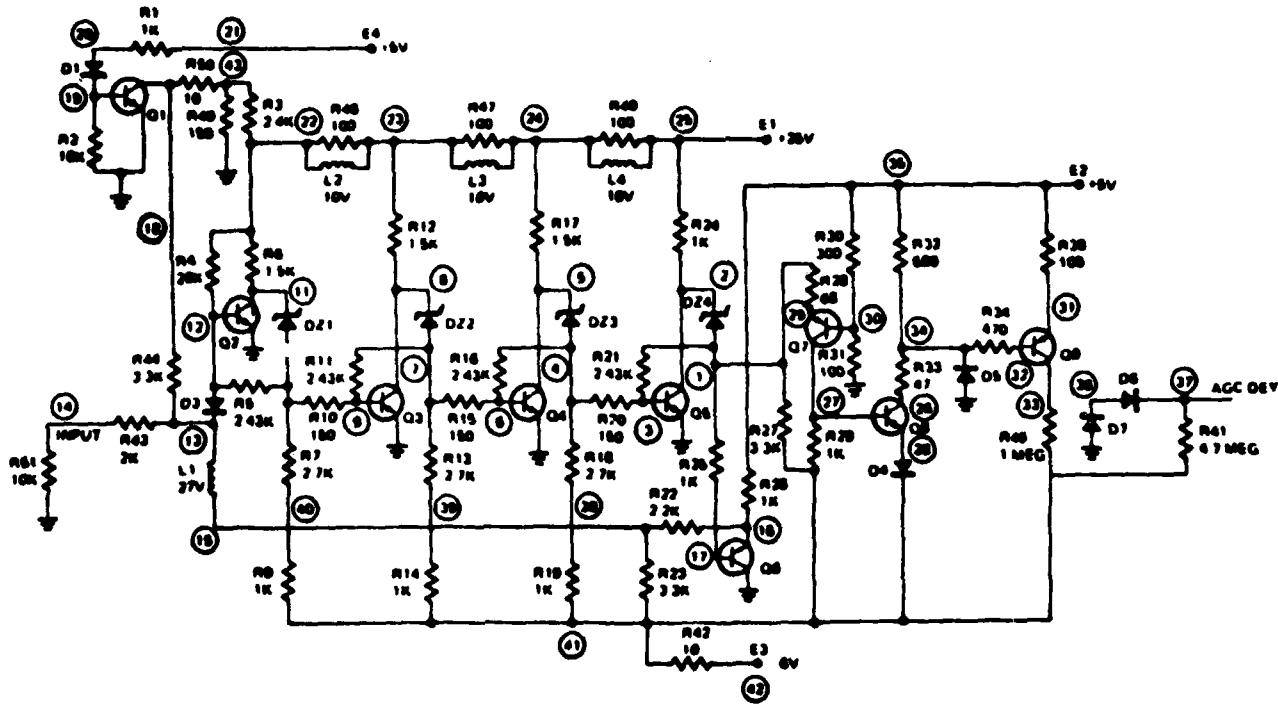


Figure 2.1. Video amplifier example used in [4].

If the catastrophic faults of *all* components, single faults as well as multiple faults are considered, the number of combinations will be enormous. Thus as the very first step, the engineer must come up with a reasonable fault list, based on his experience and the past failure history of the circuit. Strictly for the purpose of illustration, a total of 20 faults have been selected in [4], all being associated with semiconductor devices, as they are more likely to go wrong than passive components. Adopting the abbreviations B-base, C-collector, E-emitter, O-open, S-short, we have the following fault list (fault number-fault condition):

1-Q1BES, 2-Q2CES, 3-Q2BO, 4-Q3BES, 5-Q3BO, 6-Q4BES, 7-Q4BO, 8-Q5BES, 9-Q5BO, 10-Q6BES, 11-Q6BCS, 12-Q6BO, 13-DZ1O, 14-DZ1S, 15-DZ2O, 16-DZ2S, 17-DZ3O, 18-DZ3S, 19-DZ4O, 20-D24S.

The next step is to select test nodes among the accessible nodes. Discrete-component circuits have most nodes accessible. Next in line are the printed circuit boards. Finally, the integrated circuit chips have the least number of nodes accessible for testing. Even if the accessibility of nodes presents no problem, it is highly important to use as few test nodes as possible, in view of the cost of making test node connections.

In the absence of any better guidelines one may *initially* choose all or most of the accessible nodes as the test nodes. Later on, after circuit simulations under all fault conditions have been carried out, a suitable fault isolation algorithm may be executed to reduce the number of test nodes and yet maintains a satisfactory fault isolation.

In the present video amplifier circuit, 10 test nodes are chosen initially. They are nodes 2, 5, 8, 11, 16, 18, 26, 27, 33, and 36, again all associated with semiconductor devices.

In general, the nominal and fault circuits must be simulated under more than one *input combinations* in order to achieve adequate separation of faults. Each input combination is also called an *input vector*. At present, no algorithm is available for the selection of the input vectors. Trial and error, guided by knowledge of the particular circuit, seems to be the only avenue at this time. The selection of stimuli is initially made by the engineer and expanded if detection (separation from the nominal case) and isolation (separation among fault cases) requirements are not met. In [4], the following fault detection criterion is used. A fault is considered detectable only if for some input vector

$$\begin{aligned} & \sum [(\text{calculated test node voltage, fault}) \\ & - (\text{calculated test node voltage, nominal})]^2 \\ & \geq 0.5 \times (\text{number of test nodes}) \end{aligned}$$

where the summation is over all test nodes. This criterion stems from the assumption

that an average voltage deviation of 0.7 V (a diode drop) at each node is reasonable for detection. Therefore, for N test nodes, $\Sigma \Delta V^2 \geq (0.7)^2 N \simeq 0.5N$.

For the video amplifier of Fig. 1, two input vectors have been used. If we let

$$V_{IN} = (V_{14}, V_{25}, V_{35}, V_{42}, V_{21})$$

then the two chosen input vectors are (in volts):

$$V_{IN1} = (30, 25, 5, -6, 5)$$

and

$$V_{IN2} = (-30, 25, 5, -6, 5)$$

Note that the last four entries in the input vectors are simply the dc power supply voltages for the amplifier. Only the first entry, ± 30 volts, represents the externally applied dc input.

As there are 21 circuit conditions (1 nominal and 20 faulty) and 2 input vectors, a total of $21 \times 2 = 42$ circuit simulations must be carried out to determine the 10 selected test node voltages. In [4], the program SYSCAP is used. Of course, one can use any other suitable CAD programs such as SPICE2[9], ASTAP[10], etc. The data to be gathered from these simulations will be $21 \times 2 \times 10 = 420$ voltage values. These values will be processed by the fault isolation algorithm described below to eliminate the unnecessary test nodes. The voltage values corresponding to the retained test nodes then constitute the fault dictionary.

2.2 Ambiguity Sets, Fault Isolation, and Fault Dictionary

Before describing the fault isolation algorithm, it is necessary to introduce the concept of an *ambiguity set*. Consider a hypothetical case of a circuit with 2 test nodes and 6 faults. Suppose that the use of a CAD program on this circuit yields the following results:

	NOM	F1	F2	F3	F4	F5	F6
V1	5.5	9.0	6.8	6.4	6.6	5.1	2.0
V2	4.0	8.0	5.0	5.2	7.8	7.6	3.0

These voltage values are obtained under the assumption of exact element values. In reality, the value of any element may vary within some tolerance range. For example, the resistance of a 5% 1-K Ω resistor may be anywhere from 950 to 1050 ohms. Even less certain do we know about the parameter values of semiconductor devices. This, coupled with unavoidable measurement errors force us to be more cautious in judging the circuit conditions. In the present hypothetical example, if the *measured* value of

V_1 is 5.4 volts we really cannot be certain whether the circuit is under nominal, or fault #5 condition. All we can say is that when $V_1 = 5.4$ volts, the circuit is under either nominal or fault #5 condition. Thus in this case, NOM and F5 form what is called an *ambiguity set*. Following [4], we may reasonably define the voltage of an ambiguity set to have a range of ± 0.7 volt about its center value, and that different ambiguity set voltage ranges do not overlap. When we speak of an ambiguity set, we refer to the circuit conditions that produce voltages within the same ambiguity set voltage range. In the present example, test node 1 has 4 ambiguity sets and test node 2 has 3 ambiguity sets which are listed below.

(node, ambiguity set)	circuit condition	voltage range
(1,1)	NOM, F5	4.6 - 6.0
(1,2)	F2, F3, F4	5.7 - 7.1
(1,3)	F1	8.3 - 9.7
(1,4)	F6	1.3 - 2.7
(2,1)	NOM, F6	2.8 - 4.2
(2,2)	F1, F4, F5	7.1 - 8.5
(2,3)	F2, F3	4.4 - 5.8

An examination of the above table shows that fault F4 cannot be isolated from F2 and F3 if we use test node 1 only. Similarly, F4 cannot be isolated from F1 and F5 if we use test node 2 only. However, if both test nodes 1 and 2 are used then F4 can be isolated. This is because that F4 is only fault that occurs simultaneously in ambiguity sets (1,2) and (2,2). On the other hand faults F2 and F3 cannot be isolated even if both test nodes are used.

In order to obtain maximum isolation, the ambiguity sets must be manipulated to determine which faults can be isolated and what nodes will provide the greatest degree of isolation. In [4], two ground rules are used:

Rule 1. Any ambiguity set which has a single fault within it, uniquely defines that fault at that test node.

Rule 2. Ambiguity sets whose intersection (the fault is the only common element in all sets) or symmetric difference (the fault is the only different element and it is contained in only one set) result in a single fault also uniquely define the fault. In this case, the test nodes for each ambiguity set is required.

Consider again the above hypothetical example. F1 and F6 each can be uniquely defined by measuring voltage V_1 only, according to Rule 1. NOM (nominal case) and F5 each can be isolated by measuring both V_1 and V_2 , according to Rule 2, as we have

$$\begin{aligned}\text{NOM} &= \{\text{amb.set}(1,1)\} \cap \{\text{amb.set}(2,1)\} \\ &= \{\text{NOM}, \text{F5}\} \cap \{\text{NOM}, \text{F6}\}\end{aligned}$$

and

$$\begin{aligned}\text{F5} &= \{\text{amb.set}(1,1)\} \cap \{\text{amb.set}(2,2)\} \\ &= \{\text{NOM}, \text{F5}\} \cap \{\text{F1}, \text{F4}, \text{F5}\}\end{aligned}$$

Fault F4 can be isolated by either Rule 1 or Rule 2, as we have

$$\begin{aligned}\text{F4} &= \{\text{amb.set}(1,2)\} \cap \{\text{amb.set}(2,2)\} \\ &= \{\text{F2}, \text{F3}, \text{F4}\} \cap \{\text{F1}, \text{F4}, \text{F5}\}\end{aligned}$$

or

$$\begin{aligned}\text{F4} &= \text{sym. difference of } \{\text{amb.set}(1,2)\} \text{ and } \{\text{amb.set}(2,3)\} \\ &= \text{sym. difference of } \{\text{F2}, \text{F3}, \text{F4}\} \text{ and } \{\text{F2}, \text{F3}\}\end{aligned}$$

Having described the concept of ambiguity sets and the rules for their manipulations, let us return to the consideration of the video amplifier circuit of Figure 2.1. At each of ten test nodes and for each of the input vectors, the ambiguity sets are determined. For node 11, the results are as follows (0 stands for nominal case):

For input $V_{14} = 30$ V, two ambiguity sets,

$$\begin{aligned}(11,1)_1 &= \{2, 6, 7, 15, 19\} \\ (11,2)_1 &= \{0, 1, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 20\}\end{aligned}$$

For input $V_{14} = -30$ V, five ambiguity sets,

$$\begin{aligned}(11,1)_2 &= \{2\} \\ (11,2)_2 &= \{5\} \\ (11,3)_2 &= \{13\} \\ (11,4)_2 &= \{14\} \\ (11,5)_2 \text{ lieup} &= \{0, 1, 3, 4, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20\}\end{aligned}$$

The complete tabulation of the ambiguity sets for all nodes is given as Table 1 in [4].

After applying the two rules described above to these ambiguity sets, it is found that only 5 test nodes (nodes 11, 8, 2, 5 and 16) are needed for the isolation of 19 out of the 20 preselected faults. The two faults that can not be uniquely identified are faults 10 and 12. In general, to achieve a higher degree of isolation, we have to use more test

nodes, or more input vectors. This is, however, unnecessary in the present case since faults 10 and 12 are on the same replaceable component, transistor Q6. If either fault 10 or fault 12 has occurred then transistor Q6 has to be replaced anyway. Further isolation between faults 10 and 12 is therefore unnecessary.

The Fault Dictionary now consists of the dc nodal voltages at the five test nodes for two inputs under each of the 21 circuit conditions (1 nominal and 20 faulty circuits). This represents a total of 210 (5 x 2 x 21) voltage values to be stored in the fault dictionary on the ATE. A portion of the fault dictionary, corresponding to faults 4, 10, 14 and the nominal circuit is given below.

Fault	input V14	V11	V8	V5	V2	V16
F4	-30	5.32	5.99	0.09	6.95	0.02
	30	0.15	5.99	0.09	6.95	0.03
F10	-30	5.93	0.09	5.93	0.12	3.07
	30	0.20	5.97	0.09	6.91	4.22
F14	-30	1.70	0.08	5.92	0.12	3.11
	30	0.13	6.00	0.09	6.95	4.03
NOM	-30	5.90	0.08	5.95	0.12	3.09
	30	0.18	5.95	0.09	6.93	4.10

This completes the pretest setting up of the fault dictionary.

2.3 Post-test Analysis

Now suppose that a faulty circuit has been tested, and the following measured data have been obtained:

input V14	V11	V8	V5	V2	V16
-30	5.38	6.45	0.06	6.73	0.10
+30	0.15	6.46	0.06	6.77	0.13

We shall illustrate the post-test calculations required to isolate the fault. Let $SSD(FJ)$ denote the sum of square-deviations corresponding to fault FJ , i.e.,

$SSD(FJ) \triangleq \sum (\text{measured node voltage} - \text{calculated node voltage for fault } FJ)^2$

where the summation is over all test nodes and all input vectors. Then after $SSD(FJ)$ have been calculated for all faults, the one leading to the smallest value is considered to be the fault that has occurred. For the present case, we have (from the previous tables):

$$\begin{aligned}
 SSD(F4) = & [(1.82 - 5.32)^2 + (0.07 - 5.99)^2 + (6.38 - 0.09)^2 \\
 & + (0.09 - 6.95)^2 + (2.31 - 0.02)^2]
 \end{aligned}$$

$$\begin{aligned} & + [(0.11-0.15)^2 + (6.46-5.99)^2 + (0.07-0.09)^2] \\ & + (6.73-6.95)^2 + (0.13-0.03)^2] \\ & = 139.16 + 0.28 \approx 139 \end{aligned}$$

Similarly, using the data shown in the two previous tables, we find

$$\text{SSD}(F10) = 35$$

$$\text{SSD}(F14) = 1.2$$

$$\text{SSD}(\text{NOM}) = 34$$

If the SSD values for all other faults are also calculated (data not shown in [4]), the final result is as follows (Table 3 of [4]), third row:

Fault	1	2	3	4	5	6	7	8	9	10
SSD	18	129	189	139	167	120	117	70	74	35

Fault	11	12	13	14	15	16	17	18	19	20
SSD	21	35	300	1.2	196	40	143	40	104	36

Since the minimum value of SSD is 1.2, and occurs for $FJ = 14$, we conclude that the measured data indicates the occurrence of fault 14.

This completes the description of an existing fault dictionary approach [4].

3. PROPOSED NEW APPROACHES

The dc fault dictionary approach [4] described in the previous section has been successful for medium size circuits, e.g. the video amplifier circuit in [4]. When larger circuits are considered, the first difficulty encountered is the enormous amount of pre-test computation. For every possible hard failure, a dc analysis of the nonlinear circuit must be performed, using a circuit simulator such as SPICE2 [9], SYSCAP [4], or ASTAP [10]. Assume that the fault list consists of 200 preselected faults (single or simultaneous). Then 200 dc analyses must be done. All existing circuit simulators use Newton iterative procedure [11] (or some of its modified forms) to do the dc analyses. These 200 analyses will be done separately. There is no way to utilize the result of any one analysis in another.

Our aim in the pre-test stage is to find a new way of performing dc analysis which is faster than the use of Newton algorithm, possibly at some loss of the accuracy of calculated node voltages. Furthermore, this new method should have the desired property that the result of one analysis can be used to expedite the analysis of the next fault

circuit.

In the existing dc fault dictionary approach [4], the post-test analysis consists of the calculations of many SSD(sum of squared deviations). If there are N test nodes, F fault conditions, and M input vectors, then after the test of a faulty circuits has been made, we have to perform

$$N \times M \times (F+1) \text{ multiplications}$$

and

$$(N \times M - 1) \times (F+1) \text{ additions/subtractions}$$

in order to identify the fault. For the video amplifier of Section 2, where $N = 5$, $M = 2$, and $F = 20$, the number of post-test arithmetic operations required to identify the fault are 210 multiplications and 189 additions/subtractions.

Our aim in the post-test stage is to find a new method such that the identification of fault requires practically no arithmetic operations. A mere dictionary "look up" operation should be enough to identify the fault. This requirement is motivated by two practical considerations. (1) With such a simple post-test procedure, the dc dictionary approach can be used in the field (on board a ship, for example) by a technician, as all he has to do is to make measurements and then loop up the fault dictionary. No computation is needed. (2) Hardware can be constructed with a microprocessor chip and relatively simple decision making circuitry such that when the measurement data are received, the fault condition of the circuit will be displayed automatically, eliminating even the human operation of fault dictionary look-up.

To achieve the two aims stated above, we have attacked the problem with techniques quite different from those in [4]. The following is a brief summary of the distinct features of our new approaches.

(1) Model all nonlinearities by piecewise linear characteristics [12].

(2) Use switches (instead resistances as in [7] and several other papers) to represent open circuits and short circuits.

(3) Solve the piecewise linear resistive network containing switches by the use n-port theory [13] and Lemke's complementary pivot algorithm [14].

(4) Use multi-level logic to reduce the number of test nodes and to generate integer fault codes for the fault dictionary.

These will now be explained in detail in Sections 4 to 8.

4. MODELING OF PIECEWISE LINEAR RESISTIVE NETWORKS WITH IDEAL DIODES

4.1 Piecewise Linear Approximation to a Smooth Curve.

Nonlinear characteristics encountered in dc analysis of a circuit are usually associated with semiconductor devices, and are usually in the form of polynomial, exponential or some special functions. Some typical examples are:

Junction diode

$$I_D = I_s(e^{\theta V_D} - 1)$$

Bipolar junction transistor [15, p. 13]

$$I_E = -I_{ES}(e^{\theta V_{BE}} - 1) + \alpha_R I_{CS}(e^{\theta V_{BC}} - 1)$$
$$I_C = -I_{CS}(e^{\theta V_{BC}} - 1) + \alpha_F I_{ES}(e^{\theta V_{BE}} - 1)$$

As is well known, a nonlinear curve can be approximated by a piecewise linear (hereafter abbreviated PWL) curve to any desired accuracy by increasing the number of straight line segments, and hence the number of break points. Figure 4.1 shows a 5-segment approximation to the diode exponential I-V curve. In many applications, including the present case of dc fault analysis, a 3-segment, or even a 2-segment approximation to the diode I-V curve is adequate.

Of course nonlinear characteristics need not occur in 2-terminal elements only. The bipolar transistor whose model equations are given above is an example. However, it has been clearly demonstrated in [12] that models for nonlinear circuits can be constructed in such a way that all nonlinearities are associated with 2-terminal nonlinear resistors. It might be surprising to know that even nonlinear capacitors and inductors can be modeled with nonlinear resistors (plus linear elements, of course). In this research, we assume that all nonlinearities are associated with 2-terminal resistors.

Before the days of digital computers, the PWL approach was considered a very powerful means for understanding the behavior of electronic circuits, and for analyzing the same manually. In fact, there were undergraduate textbooks in electronic circuits that used exclusively the PWL approach [12, 16], despite its obvious drawback of being less accurate.

With the advent of high speed digital computers, and with the demand for more accurate analysis result, the PWL approach has given in to the Newton iterative approach [13], because the exact nonlinear characteristics are used in the latter. Because of the repeated evaluation of the Jacobian matrix, Newton procedure is in general more time-consuming than the PWL approach. Here we see another example of the trade-off between speed and accuracy in computer-aided circuit analysis.

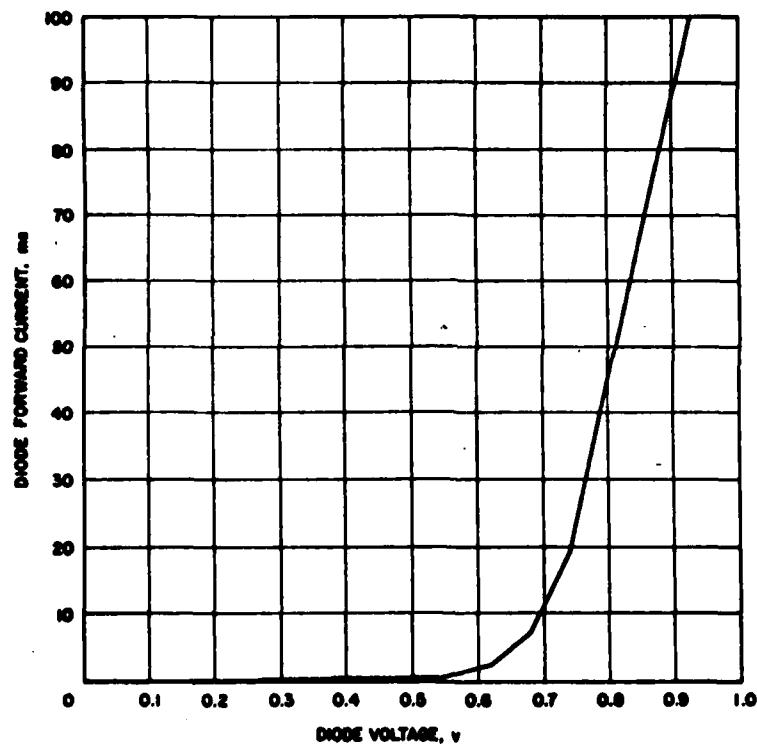


Figure 4.1 A 5-segment approximation to a diode I-V curve.

In our present research of fault diagnosis, we have a situation where the demand of accuracy is less severe than the usual circuit simulation. This is because of the safety margins provided by the ambiguity sets. To illustrate this point, consider the video amplifier of Section 2. Under fault #5, the exact voltage of node 1 is $V_1 = 5.1$, and F5 belongs to ambiguity set (1,1) which has a range from 4.6 to 6.0 volts. Now suppose that due to model inaccuracies or the limited accuracy of the computer, the calculated value of V_1 becomes 5.5 volts (8% relative error). No harm is done! F5 still belongs to ambiguity set (1,1), and the fault isolation process goes on just as before. It is important, however, to avoid very large errors that might change the constituents of the ambiguity sets.

In passing it should be pointed out that although statistical variations of element values have not been considered in our present research, the use of ambiguity sets does provide considerable safeguard against erroneous fault identification due to the variation of element values within their tolerance margins.

4.2 A Simple PWL Resistive Network and Its Solution

Once the characteristic of all nonlinear resistors have been approximated by PWL segments, they can be modeled with ideal diodes, dc independent voltage sources and current sources. An ideal diode here is defined as a 2-terminal element that behaves as an open circuit when reverse biased and as a short circuit when forward biased. Figure 4.2 shows the modeling of four simple PWL I-V curves using ideal diodes [16]. The technique for modeling the most general PWL I-V curves (negative slope segments allowed) may be found in [12]. Usually, but not always, the number of break joints in the I-V curve is equal to the number of ideal diodes in the circuit model.

A nonlinear network without energy storage elements, and with all nonlinearities in the form of piecewise linear curve is called a PWL resistive network. From the above discussions, we see that any PWL resistive network can be modeled with linear resistors, linear controlled source and ideal diodes. We shall demonstrate the formation of a PWL resistive network and its solution with a simple example.

Fig. 4.3(a) shows a single stage transistor amplifier. The input and output characteristics of the transistor are assumed to be PWL and are given in Fig. 4.3(b). From these characteristics, we can easily construct the PWL model for the transistor [12, Chapter 11]. The result is shown in Fig. 4.3(c).

Now suppose that we wish to find the dc operating point of the amplifier. In dc steady state, capacitances behaves as open circuits. After removing the two capacitances in Fig. 4.3(a), and replacing the transistor by its PWL model of Fig. 4.3(c), we obtain a PWL resistive network as shown in Fig. 4.3(d).

Many methods are available for solving the PWL network of Fig. 4.3(d). To be consistent with our later development we shall use the n-port approach [13, Chapter 6].

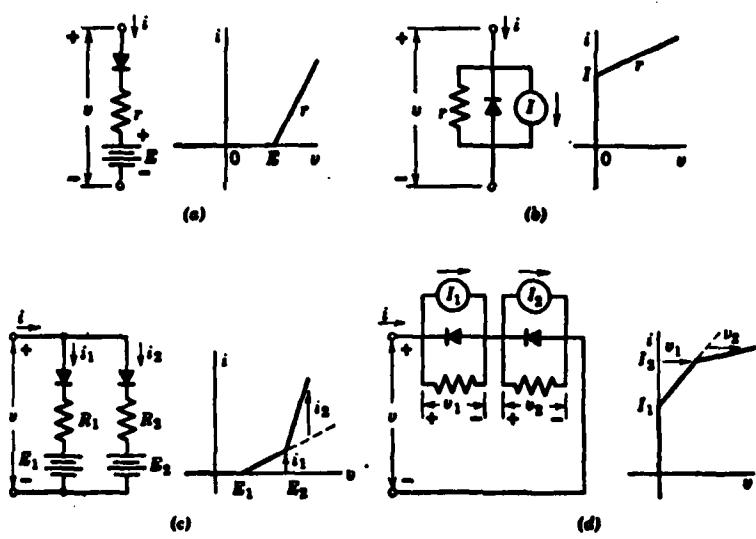
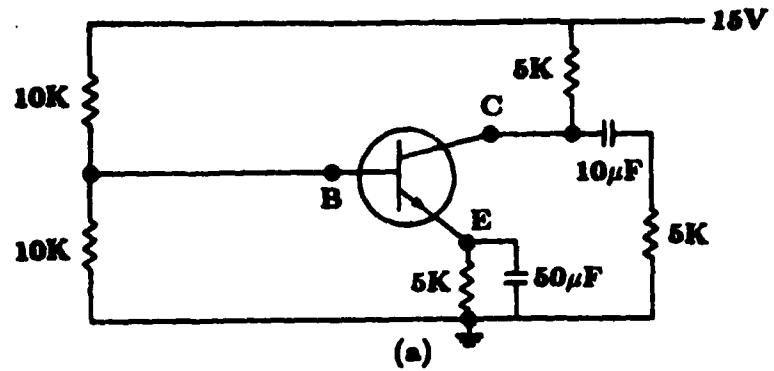
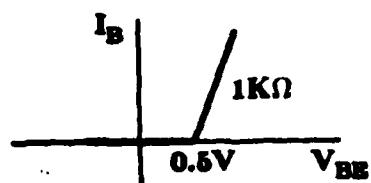


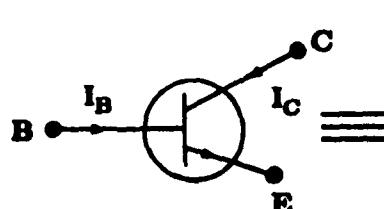
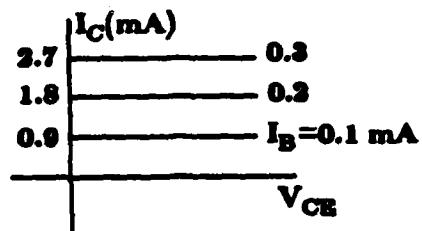
Figure 4.2 Modeling PWL curves with ideal diodes.



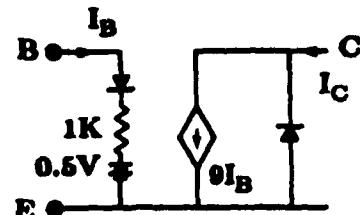
(a)



(b)



(c)



(d)

Figure 4.3 A transistor amplifier and its piecewise linear model.

Suppose that we extract the two ideal diodes to form a 2-port. The elements inside this 2-port are dc independent sources, linear resistors, and controlled sources. By a straightforward circuit analysis, we find the following constraint equation on the port voltages and currents:

$$\begin{bmatrix} V_{d1} \\ V_{d2} \end{bmatrix} = \begin{bmatrix} 56 & -5 \\ -95 & 10 \end{bmatrix} \begin{bmatrix} I_{d1} \\ I_{d2} \end{bmatrix} + \begin{bmatrix} -7 \\ 15 \end{bmatrix} \quad (4.1)$$

where I_d denotes the diode forward current, and V_d denotes the diode reverse bias voltage, both being nonnegative. The two ideal diodes may be viewed as the terminations for the two ports. Therefore, the ideal diode characteristics form additional constraints on the port voltages and currents as given below

$$\begin{bmatrix} V_{d1} \\ V_{d2} \end{bmatrix} \geq 0, \quad \begin{bmatrix} I_{d1} \\ I_{d2} \end{bmatrix} \geq 0 \quad (4.2)$$

$$V_{d1} I_{d1} = 0, \quad V_{d2} I_{d2} = 0 \quad (4.3)$$

The problem now is to find $(V_{d1}, V_{d2}, I_{d1}, I_{d2})$ that will satisfy equations (4.1) (4.2) and (4.3), which is a special case of the so-called complementary problem to be discussed in Section 5.

For the present simple case, we can use the exhaustive search method. Note that (4.2) and (4.3) together imply that at least one element in any complementary pair (V_{dj}, I_{dj}) , $j = 1, 2$, must be zero. Thus there are only 4 cases to be considered:

- Case 1. $I_{d1} = 0, I_{d2} = 0$
- Case 2. $I_{d1} = 0, V_{d2} = 0$
- Case 3. $V_{d1} = 0, I_{d2} = 0$
- Case 4. $V_{d1} = 0, V_{d2} = 0$

For Case 1, we let $I_{d1} = 0$ and $I_{d2} = 0$ in Eq. (4.1). This immediately leads to $V_{d1} = -7, V_{d2} = 15$. The negative value of V_{d1} violates condition (4.2). Therefore Case 1 does not yield a solution to the problem.

Next consider Case 2, $I_{d1} = 0, V_{d2} = 0$. Following the procedure in Case 1, we will first express V_{d1} and I_{d2} in terms of I_{d1} and V_{d2} . In network sense, this amounts to finding another representation of the 2-port. The goal can be achieved by simple matrix manipulations. In Eq. (4.1), we consider V_{d1} and I_{d2} as unknowns and solve for them. The result is:

$$\begin{bmatrix} V_{d1} \\ I_{d2} \end{bmatrix} = \begin{bmatrix} 8.5 & -0.5 \\ -9.5 & -0.1 \end{bmatrix} \begin{bmatrix} I_{d1} \\ V_{d2} \end{bmatrix} + \begin{bmatrix} 0.5 \\ -1.5 \end{bmatrix} \quad (4.4)$$

Setting $I_{d1} = 0, V_{d2} = 0$ in Eq. (4.4) we immediately obtain $V_{d1} = 0.5, I_{d2} = -1.5 < 0$. Since I_{d2} is negative, Case 2 does not yield a solution either.

Next consider Case 3, $V_{d1} = 0, I_{d2} = 0$. Following the similar procedure, we first obtain

$$\begin{bmatrix} I_{d1} \\ V_{d2} \end{bmatrix} = \begin{bmatrix} 0.018 & 0.089 \\ -1.696 & 1.518 \end{bmatrix} \begin{bmatrix} V_{d1} \\ I_{d2} \end{bmatrix} + \begin{bmatrix} 0.125 \\ 3.125 \end{bmatrix} \quad (4.5)$$

Setting $V_{d1} = 0, I_{d2} = 0$, we immediately obtain

$$\begin{bmatrix} I_{d1} \\ V_{d2} \end{bmatrix} = \begin{bmatrix} 0.125 \\ 3.125 \end{bmatrix} \geq 0$$

Since all constraints have been met, we have *one valid solution*: $V_{d1} = 0, I_{d1} = 0.125, V_{d2} = 3.125, I_{d2} = 0$. From these results, the operating point of the transistor is easily found to be:

$$V_{BE} = 0.625V \quad I_B = 0.125 \text{ mA}$$

$$V_{CE} = 3.125V \quad I_C = 1.125 \text{ mA}$$

In a simple circuit such as the present one, we know that the circuit has a unique solution. Since we have found one solution in Case 3, there is no need to proceed to Case 4. However, in general, when the possibility of multiple solutions exists we have to consider all possible cases.

In the previous example of 2 ideal diodes, we obtain a solution in Case 3. But we can not expect to have such luck all the time. The successful case could have been Case 4. In general, if there are n ideal diodes, we might have to investigate 2^n cases before a solution is found. Obviously then, the exhaustive search method is practical only for PWL networks having very few ideal diodes. Efficient solution of PWL networks containing many ideal diodes (or equivalently, many straight line segments in the PWL characteristics) has been a topic of continued research interest (see [13, Chapter 7 and the references on page 323]. In the next section, we shall describe one method, the complementary pivot algorithm, which is particularly suited to our present goal of fault diagnosis.

5. THE COMPLEMENTARY PROBLEM AND METHODS OF SOLUTION

5.1 The Complementary Problem

In the previous section we have seen an example of the complementary problem of order 2. The general case of an n th order complementary problem [14] is to find vectors w and z such that

$$w = Mz + q \quad (5.1a)$$

$$w \geq 0, z \geq 0 \quad (5.1b)$$

$$w^t z = 0 \quad (5.1c)$$

where the matrix M is a square matrix of order n , and w, z, q are all n -vectors. M and q are given, while w and z are unknowns to be found. For example, the following is a complementary problem of order 3.

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} -2 & -14 & 4 \\ -4 & -4 & 2 \\ 4 & 4 & -2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} -4 \\ -6 \\ 10 \end{bmatrix} \quad (5.2a)$$

$$w \geq 0, z \geq 0 \quad (5.2b)$$

$$w^t z = w_1 z_1 + w_2 z_2 + w_3 z_3 = 0 \quad (5.2c)$$

If $q \geq 0$, then obviously one solution to the complementary problem (5.1) is $w = q$ and $z = 0$. But this may not be the only solution. In fact, a complementary problem may have no solution, a unique solution, several solutions, or infinitely many solutions. To see these possibilities, we need only consider the following first order complementary problem.

Example 5.1

$$w = mz + q,$$

$$w \geq 0, z \geq 0, wz = 0$$

Case 1. $w = z + 1$.

There is a unique solution: $w = 1, z = 0$

Case 2. $w = -z + 1$.

There are two solutions: $w = 1, z = 0$ and $w = 0, z = 1$.

Case 3. $w = -z - 1$. There is no solution.

Case 4. $w = 0z + 0$. There are infinitely many solutions: $w = 0, 0 \leq z < \infty$.

To find all solutions to a complementary problem is known to be extremely time-consuming. In many practical applications, we either know that there can be at most

one solution, or are content with obtaining any one solution. In such case, the complementary pivot method, to be described in Section 5.3, has been found very powerful. Before discussing the complementary pivot method, it is very instructive to consider another method called the exhaustive search method, which has a clear relationship with n-port network theory.

5.2 The Exhaustive Search Method

The exhaustive search method essentially reduces the original complementary problem to a number of problems each of which amounts to finding n unknowns from n linear simultaneous equations. To see how this is achieved let us review the constraints (5.1b) and (5.1c). These two constraints together imply that in every pair of unknowns (w_i, z_i) , called a *complementary pair*, at least one element is zero (or equivalently, $w_i z_i = 0$). Since there are n complementary pairs, any solution to (5.1) must have n unknowns *set to zero* (some other unknowns may be *found to be zero subsequently*). The n unknowns to be set to zero may be chosen arbitrarily, the only restriction being that they should not contain both elements of a complementary pair. After setting n unknowns in Eq. (5.1a) to zero, we obtain a set of n equations in the remaining n unknowns, which may then be solved by any of the well-known linear equation solution methods. If all of the answers are nonnegative, then constraints (5.2b) have been met, and a solution to the complementary problem has been found. Otherwise, set a different group n unknowns to zero and repeat the procedure, until a correct group has been selected. Now we may view Eq. (5.1a) as a constraint equation for a linear resistive n -port containing dc independent sources as follows [13, Chapter 6]: \mathbf{z} and \mathbf{w} are the independent and dependent port variables, respectively, and \mathbf{q} is the *forcing vector* (due to independent sources inside the n -port). Furthermore, if z_i is the current into (voltage across) the i th port, then w_i is the voltage across (current into) the i th port. With such an interpretation of Eq. (5.1a), it is seen that solving a complementary problem by the exhaustive search method is the same as finding a port-classification for an n -port such that all elements in the forcing vector of the hybrid equation [13] are non-negative.

To facilitate the solution process, it is more convenient to rewrite Eq. (5.1a) as

$$[1 - \mathbf{M}] \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix} = \mathbf{q} \quad (5.3)$$

Further simplification in bookkeeping results if we display the information contained in (5.3) in a tableau form as follows

$$\left[\begin{array}{c|c|c|c} w_1 \dots w_n & z_1 \dots z_n & & \\ \hline 1 & -\mathbf{M} & & \mathbf{q} \end{array} \right] \quad (5.4)$$

Note that in the tableau (5.4) we have used the unknowns $w_1, \dots, w_n, z_1, \dots, z_n$ to label the columns with which they are associated. In this initial tableau, and in all later modified tableaus, we always arrange the columns in such a manner that the second block contains those unknowns to be set to zero while the first block contains those unknowns to be solved for.

To illustrate the exhaustive search procedure, let us consider the complementary problem given by (5.2). The tableau corresponding to Eq. (5.2a) is given below.

$$\left[\begin{array}{ccc|ccc|c} w_1 & w_2 & w_3 & z_1 & z_2 & z_3 & q \\ 1 & 0 & 0 & 2 & 14 & -4 & -4 \\ 0 & 1 & 0 & 4 & 4 & -2 & -6 \\ 0 & 0 & 1 & -4 & -4 & 2 & 10 \end{array} \right] \quad (5.5)$$

Setting the unknowns in the second block to zero, i.e., $z_1 = z_2 = z_3 = 0$, we immediately obtain a solution to (5.2a): $w_1 = -4$, $w_2 = -6$, $w_3 = 10$. Since the nonnegativity constraint (5.2b) is violated, this solution to (5.2b) is not a solution to the complementary problem (5.2).

The exhaustive search method now proceeds with a different choice of unknowns to be set to zero. Suppose that z_1 and w_1 columns are switched in the tableau. We obtain the following new tableau.

$$\left[\begin{array}{ccc|ccc|c} z_1 & w_2 & w_3 & w_1 & z_2 & z_3 & q \\ 2 & 0 & 0 & 1 & 14 & -4 & -4 \\ 4 & 1 & 0 & 0 & 4 & -2 & -6 \\ -4 & 0 & 1 & 0 & -4 & 2 & 10 \end{array} \right] \quad (5.6)$$

We now set (w_1, z_2, z_3) the zero and solve for (z_1, w_2, w_3) . This amounts to "pivoting" on the (1,1) element (By "pivoting" on a nonzero element a_{ij} of a matrix \mathbf{A} , we mean using elementary row operations to reduce the j th column to an identity column having $a_{ij} = 1$). The aim of pivoting is to reduce the first block to an identity matrix. The result is

$$\left[\begin{array}{ccc|ccc|c} z_1 & w_2 & w_3 & w_1 & z_2 & z_3 & q \\ 1 & 0 & 0 & 0.5 & 7 & -2 & -2 \\ 0 & 1 & 0 & -2 & -24 & 6 & 2 \\ 0 & 0 & 1 & 2 & 24 & -6 & 2 \end{array} \right] \quad (5.7)$$

Because the q -vector (rightmost columns), contain a negative element, setting (w_1, z_2, z_3) to zero does not yield a solution to (5.2) either. We must continue the trials.

Suppose that in the initial tableau (5.5), w_3 and z_3 are switched. We have

$$\left[\begin{array}{cccc|cc|c} w_1 & w_2 & z_3 & z_1 & z_2 & w_3 & q \\ 1 & 0 & -4 & 2 & 14 & 0 & -4 \\ 0 & 1 & -2 & 4 & 4 & 0 & -6 \\ 0 & 0 & 2 & -4 & -4 & 1 & 10 \end{array} \right] \quad (5.8)$$

Pivoting on (3,3) element to reduce the first block to an identity matrix we obtain

$$\left[\begin{array}{cccc|cc|c} w_1 & w_2 & z_3 & z_1 & z_2 & w_3 & q \\ 1 & 0 & 0 & -6 & -6 & 2 & 16 \\ 0 & 1 & 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 1 & -2 & -2 & 0.5 & 5 \end{array} \right] \quad (5.9)$$

Since in the above tableau $q \geq 0$, we have obtain a solution to (5.2), namely $z_1 = z_2 = w_3 = 0$, $w_1 = 16$, $w_2 = 4$, $z_3 = 5$.

The number of arithmetic operations needed in going from (5.6) to (5.7), or from (5.8) to (5.9) can be easily determined to be

12 multiplications

8 additions

For the general case of an n th order complementary problem the corresponding numbers are

$n(n+1)$ multiplications

(n^2-1) additions.

Since for each complementary pair (w_i, z_i) , either w_i or z_i can be included in the second block, there and a total of 2^n ways of selecting the unknowns for the second block. These 2^n combinations can be arranged sequentially such that each one differs from its previous one in only one variable. Therefore, to find all possible solutions, or just to find one solution, we have to perform, in the worst case, 2^n matrix reductions each requiring $n(n+1)$ multiplications and (n^2-1) additions. This exponential growth of operation counts clearly restricts the use of the exhaustive search method to very small size problems (say $n \leq 3$ for manual solution, and $n \leq 20$ for computer solution). In our present research in fault diagnosis, we can easily have $n > 100$ for realistic circuits. Clearly a much more efficient method of solving the complementary problem must be sought.

It has been shown by those in the operations research area that many optimization problems such as linear programs and convex quadratic programs can be formulated as a complementary problem [14]. Several algorithms have been proposed for solving the complementary problem. In this research project, we have chosen Lemke's

complementary pivot algorithm for solving the complementary problem, because the algorithm has been found superior to other exciting methods [14, p. 539]. A brief description of the complementary pivot algorithm follows.

5.3 Lemke's Complementary Pivot Method

In the exhaustive search method described above. We choose one combination of n variables, set them to zero, and solve for the remaining n variables. If the answers are not all nonnegative, then we choose another combination of n variables and repeat the process. There is no restriction, or guideline as to which combination of n variables should be examined next. The Lemke's complementary pivot algorithm is also a sequential search method but with a *definite* sequence of combinations of variables to be examined. The following is a brief description of the algorithm. For a more detailed discussion of this method, see [14, Chapter 11].

Lemke proposed to solve (5.1) by the introduction of an artificial variable z_0 . Consider the new system of equations

$$\mathbf{w} = \mathbf{Mz} + \mathbf{ez}_0 + \mathbf{q} \quad (5.10a)$$

$$\mathbf{w} \geq 0, \mathbf{z} \geq 0, z_0 \geq 0 \quad (5.10b)$$

$$\mathbf{w}^t \mathbf{z} = 0 \quad (5.10c)$$

where \mathbf{e} is an n -vector $[1, \dots, 1]^t$. Observe that any solution to (5.10) with $z_0 > 0$ is not a solution to (5.1), and is called an *almost complementary solution* to (5.1). However a solution to (5.10) with $z_0 = 0$ is also a solution to (5.1). In essence, Lemke's algorithm is a sequential search of the solutions to (5.10), until one solution with $z_0 = 0$ is found, which is then a solution to the complementary problem (5.1)

To facilitate the solution process, we first rewrite (5.10a) as

$$\begin{bmatrix} 1 & -\mathbf{M} & -\mathbf{e}_0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \\ z_0 \end{bmatrix} = \mathbf{q} \quad (5.11)$$

Further simplification is bookkeeping results if we display the information contained in (5.11) in a tableau form as follows:

$$\left[\begin{array}{ccc|c} \mathbf{W}_1 \cdots \mathbf{W}_n & \mathbf{Z}_1 \cdots \mathbf{Z}_n & z_0 & \\ \hline & & -1 & \\ & & 1 & \\ & & \vdots & \\ 1 & -\mathbf{M} & \vdots & \mathbf{q} \\ & & \ddots & \\ & & -1 & \end{array} \right] \quad (5.12)$$

Note that in the above tableau we have used the unknowns $w_1, \dots, w_n, z_1, \dots, z_n$ and z_0 to label the columns with which they are associated with. In this initial tableau, and in all later modified tableaus obtained by switching columns, we will maintain the following two properties for the first block:

- (1) The $n \times n$ matrix is nonsingular.
- (2) For each complementary pair (w_i, z_i) , at most one element is included in this block.

In the complementary problem literature, the variable in the first block are called *basic variables*, whereas the variables in the second block are called *nonbasic variables*. Any n basic variables constitute a *basis*. Nonbasic variables are to be set to zero, and basic variables are found from the resulting simultaneous equations.

As mentioned earlier, the case $\mathbf{q} \geq 0$ in (5.1) has an obvious solution $\mathbf{z} = 0, \mathbf{w} = \mathbf{q}$. Therefore assume in (5.10) some elements of \mathbf{q} are negative. Let q_i be the most negative element in \mathbf{q} . Switching columns z_0 with w_i , one obtains (illustrated for $i = 2$)

$$\left[\begin{array}{cccc|cc|c} w_1 & z_0 & \dots & w_n & z_1 & \dots & z_n & w_i \\ 1 & -1 & 0 & \vdots & -m_{11} & \dots & -m_{n1} & 0 & q_1 \\ 0 & \textcircled{-1} & 0 & \vdots & -m_{21} & \dots & -m_{n2} & 1 & q_2 \\ \cdot & -1 & \cdot & \vdots & \cdot & \ddots & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \vdots & \cdot & \ddots & \cdot & \cdot & \cdot \\ 0 & -1 & 1 & \vdots & -m_{n1} & \dots & -m_{nn} & 0 & q_n \end{array} \right] \quad (5.13)$$

We shall now set the second block variables to zero, and solve for the first block variables. This amounts to applying elementary row operations to (5.13) to reduce the first block to an identity matrix. Observe that the first block of (5.13), (and any later modified tableau) differs from an identity matrix in at most one column, the i th column in the present case. Therefore we need only pivot on the (i,i) element (shown circled in the tableau) and obtain the following new tableau:

$$\left[\begin{array}{cccc|cc|c} w_1 & z_0 & \dots & w_n & z_1 & \dots & z_n & w_i \\ 1 & 0 & 0 & \vdots & X & \dots & X & \hat{q}_1 \\ 0 & 1 & 0 & \vdots & X & \dots & X & \hat{q}_2 \\ \cdot & 0 & \cdot & \vdots & \cdot & \ddots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \vdots & \cdot & \ddots & \cdot & \cdot \\ 0 & 0 & 1 & \vdots & X & \dots & X & \hat{q}_n \end{array} \right] \quad (5.14)$$

where the entries in the second block (indicated by X) may be positive, negative, or zero. However, the entries in the third block (the new forcing vector $\hat{\mathbf{q}}$) are all

nonnegative. This is because we have chosen q_i to be the most negative element of q , which leads to

$$\hat{q}_i = -q_i > 0$$

$$\hat{q}_j = -q_i + q_j \geq 0 \quad \text{for all } j \neq i$$

From the tableau (5.14), we immediately obtain a solution to (5.10) by setting the second block variables to zero, and equating the first block variables with the elements of \hat{q} . Since

$$z_0 = \hat{q}_i = -q_i > 0$$

this solution of (5.10) is not yet a solution to (5.1), the given complementary problem.

As in the exhaustive search method, we shall now switch one column in the second block with one column in the first block. Or, in other words, we shall choose one variable to enter the basis, and another variable to leave the basis. The interchange of columns should be such that the new tableau readily leads to another solution of (5.10). This goal can be achieved by enforcing the following two rules:

Rule 1. The variable selected from the second block (nonbasic variables) for interchange is the *complement* of the variable which has entered the second block most recently. Suppose that the selected column is the k th.

Rule 2. The variable selected from the first block (basic variables) is determined by the "minimum ratio test" as follows:

Calculate \hat{q}_i/\hat{m}_{ik} for all $\hat{m}_{ik} > 0$. Suppose the minimum value of this ratio occurs when $i = j$, i.e.

$$\frac{\hat{q}_j}{\hat{m}_{jk}} = \min \left\{ \frac{\hat{q}_i}{\hat{m}_{ik}} \right\} \text{ for all } i, \text{ s.t. } \hat{m}_{ik} > 0$$

Then the j th column in the first block is selected for interchange.

After two columns have been switched, we apply elementary row operations to the new tableau and reduce the first block to an identity matrix, with its accompanying new second block and new forcing vector. Rule 2 guarantees that all elements of the new forcing vector will remain nonnegative, while Rule 1 guarantees that no complementary pair (w_i, z_i) appear in the first block. Then another solution to (5.10) can be obtained immediately by setting the second block variables to zero, and equating the first block (now an identity matrix) column variables to the elements of the latest forcing vector. At this stage, if the auxiliary variable z_0 is not included among the first block (basic variables), then a solution to the complementary problem (5.1) has been obtained. Otherwise, continue the column interchange process according to Rules 1

and 2.

Note 1. In carrying out Rule 2, if all elements in the k th column are nonpositive, then Lemke's algorithm terminates. The termination indicates one of two possibilities [17]: (1) The given problem (5.1) has no solution. (2) The given problem (5.1) has a solution, but Lemke's algorithm fails to find a solution with the given starting basis. Very often, by starting with a new basis (i.e., an alternative characterization of (5.1) obtained by interchanging one pair of complementary variable w_i and z_i) a solution to (5.1) may be found.

Note 2. In carrying out Rule 2, if the minimum ratio occurs for more than one value of i , then we say a tie has occurred. Reference [17] describes a method for breaking the tie and making the choice unique.

The above solution procedure for (5.1) is called the complementary pivot method. It is best illustrated with an example.

Example 5.2. Solve the complementary problem (5.2) by the complementary pivot method.

Solution: The initial tableau is

$$\left[\begin{array}{cccc|cccc|c} w_1 & w_2 & w_3 & z_1 & z_2 & z_3 & z_0 & q \\ 1 & 0 & 0 & 2 & 14 & -4 & -1 & -4 \\ 0 & 1 & 0 & 4 & 4 & -2 & -1 & -6 \\ 0 & 0 & 1 & -4 & -4 & 2 & -1 & 10 \end{array} \right] \quad (5.15)$$

Since the most negative element in q is -6 which occurs in the 2nd row, we interchange the 2nd column, namely w_2 with z_0 , the result is:

$$\left[\begin{array}{cccc|cccc|c} w_1 & z_0 & w_3 & z_1 & z_2 & z_3 & w_2 & q \\ 1 & -1 & 0 & 2 & 14 & -4 & 0 & -4 \\ 0 & -1 & 0 & 4 & 4 & -2 & 1 & -6 \\ 0 & -1 & -1 & -4 & -4 & 2 & 0 & 10 \end{array} \right] \quad (5.16)$$

Pivoting in (2, 2) element to reduce the first block to an identity matrix, we obtain a new tableau

$$\left[\begin{array}{cccc|cccc|c} w_1 & z_0 & w_3 & z_1 & z_2 & z_3 & w_2 & q \\ 1 & 0 & 0 & -2 & 10 & -2 & -1 & 2 \\ 0 & 1 & 0 & -4 & -4 & 2 & -1 & 6 \\ 0 & 0 & 1 & -8 & -8 & 4 & -1 & 16 \end{array} \right] \quad (5.17)$$

Note that in (5.17) $q \geq 0$. One almost complementary solution to (5.2) is obviously $z_1 = z_2 = z_3 = w_2 = 0$, $w_1 = 2$, $z_0 = 6$, $w_3 = 16$. Since $z_0 \neq 0$, this is not a solution to (5.2) yet.

According to Rule 1, the column in the second block selected for interchange is z_2 (since w_2 has entered this block most recently). There is one ratio to be considered, namely $2/10$, which is naturally also the minimum ratio. Since the element 10 occurs in the *first* row, according Rule 2, the *first* column, namely w_1 is chosen for interchange. We have, after interchanging z_2 and w_1 columns, and reducing the first block to an identity matrix by elementary row operations, the following new tableau:

$$\left[\begin{array}{ccccccc|c} z_2 & z_0 & w_3 & z_1 & w_1 & z_3 & w_2 & q \\ 1 & 0 & 0 & -2 & 0.1 & -0.2 & -1 & 0.2 \\ 0 & 1 & 0 & -4.8 & 0.4 & 1.2 & -1.4 & 6.8 \\ 0 & 0 & 1 & -9.6 & 0.8 & 2.4 & -1.8 & 17.6 \end{array} \right] \quad (5.18)$$

Note that $q \geq 0$ in (5.18). Since z_0 is still within the first block, we continue to apply Rules 1 and 2. Since w_1 has entered the 2nd block most recently, then z_1 is selected for interchange. But there is no positive element in the column z_1 , the algorithm terminates without giving a solution to (5.1).

As mentioned in Note 1 above, we should in this case try a new starting basis. Eq. (5.2a) may be expressed by the following equivalent system of equations:

$$\begin{bmatrix} z_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} -0.5 & -7 & 2 \\ 2 & 24 & -6 \\ -2 & -24 & 6 \end{bmatrix} \begin{bmatrix} w_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} -2 \\ 2 \\ 2 \end{bmatrix} \quad (5.19)$$

The initial tableau representing (5.19) is

$$\left[\begin{array}{ccccccc|c} z_1 & w_2 & w_3 & w_1 & z_2 & z_3 & z_0 & q \\ 1 & 0 & 0 & 0.5 & 7 & -2 & -1 & -2 \\ 0 & 1 & 0 & -2 & -24 & 6 & -1 & 2 \\ 0 & 0 & 1 & 2 & 24 & -6 & -1 & 2 \end{array} \right] \quad (5.20)$$

Since the most negative element in q is -2 and occurs in the first row, column z_0 is to be interchanged with column z_1 . The result is

$$\left[\begin{array}{ccccccc|c} z_0 & w_2 & w_3 & w_1 & z_2 & z_3 & z_1 & q \\ -1 & 0 & 0 & 0.5 & 7 & -2 & 1 & -2 \\ -1 & 1 & 0 & -2 & -24 & 6 & 0 & 2 \\ -1 & 0 & 1 & 2 & 24 & -6 & 0 & 2 \end{array} \right] \quad (5.21)$$

Pivoting on (1, 1) element to reduce the first block to an identity matrix, we obtain

$$\left[\begin{array}{ccccccc|c} z_0 & w_2 & w_3 & w_1 & z_2 & z_3 & z_1 & q \\ 1 & 0 & 0 & -0.5 & -7 & 2 & -1 & 2 \\ 0 & 1 & 0 & -2.5 & -31 & 8 & -1 & 4 \\ 0 & 0 & 1 & 1.5 & 17 & -4 & -1 & 4 \end{array} \right] \quad (5.22)$$

This tableau indicates an obvious almost complementary solution to (5.2) namely $w_1 = z_2 = z_3 = z_1 = 0$, $z_0 = 2$, $w_2 = 4$, $w_3 = 4$. Since $z_0 \neq 0$, this is not a solution to (5.2) yet.

Since z_1 has entered the second block most recently, according to Rule 1, we select column w_1 for interchange. The minimum ratio is $4/1.5$. Since 1.5 occurs in the *third* row, according to Rule 2, we select the *third* column, namely w_3 for interchange. After interchanging columns w_3 and w_1 , and reducing the first block to an identity matrix by row operations, we obtain

$$\left[\begin{array}{cccc|cccc|c} z_0 & w_2 & w_1 & w_3 & z_2 & z_3 & z_1 & q \\ 1 & 0 & 0 & 1/3 & 4/3 & 2/3 & -4/3 & 10/3 \\ 0 & 1 & 0 & 5/3 & 8/3 & 4/3 & -8/3 & 32/3 \\ 0 & 0 & 1 & 2/3 & 34/3 & -8/3 & -2/3 & 8/3 \end{array} \right] \quad (5.23)$$

Since z_0 is still among the basic variables, we continue the column interchange process, Rule 1 now dictates column z_3 be selected for interchange, since w_3 has entered the second block most recently. The minimum ratio among

$$\frac{10}{3} : \frac{2}{3} = 5$$

and

$$\frac{32}{3} : \frac{4}{3} = 8$$

is 5. Since the element $2/3$ occurs in the *first* row, Rule 2 dictates that the *first* column, namely z_0 be selected for interchange. After interchanging columns z_3 and z_0 , and reducing the first block to an identity matrix, we obtain

$$\left[\begin{array}{cccc|cccc|c} z_3 & w_2 & w_1 & w_3 & z_2 & z_0 & z_1 & q \\ 1 & 0 & 0 & 0.5 & -2.0 & 1.5 & -2 & 5 \\ 0 & 1 & 0 & 1 & 0 & -2 & 0 & 4 \\ 0 & 0 & 1 & 2 & 6 & 4 & -6 & 16 \end{array} \right] \quad (5.24)$$

In this tableau, the artificial variable z_0 is not among the basic variables, and the forcing vector has all nonnegative elements. Therefore we have obtained one solution to the given complementary problem (5.2), namely

$$w_3 = z_2 = z_1 = 0 \quad (z_0 = 0)$$

$$z_3 = 5, w_2 = 4, w_1 = 16$$

The solution, of course, agrees with that obtained in Sec. 5.2 by the exhaustive search method.

6. PRETEST DC ANALYSIS USING COMPLEMENTARY PIVOT METHOD

As mentioned in Section 2, the dc fault dictionary approach consists of two distinct stages: the pretest analysis and the post-test analysis. In the pretest analysis stage, the basic problem is the determination of dc node voltages under various fault conditions and dc input stimuli. Usually this is done with the aid of some circuit simulation program, such as SPICE2 [9], SYSCAP [4], ASTAP [10], etc.

We shall now present a new approach to the pretest analysis, making use of the techniques described in Section 4 and 5 as well as the multiport formulation algorithms [13, Chapter 6]. We assume that the test nodes, the input stimuli, and the list of catastrophic failure are given. Our only concern in this section is the rapid determination of test node voltages under various circuit condition.

6.1 The Use of Switches

By catastrophic failures we refer to open circuits and short circuits. One method to represent these two conditions is by way of the element value. Suppose that a resistance R_j has a nominal value of 1000 ohms. Then $R_j = \infty$ indicates an open circuit and $R_j = 0$ indicates a short circuit.

In our new approach, however, we use switches to represent open circuits and short circuits. Refer to Figure 6.1. The switch S_1 is normally closed (N.C.). Opening S_1 signifies that R_1 becomes an open circuit. The switch S_2 is normally open (N.O.). Closing S_2 signifies that R_2 becomes a short circuit. A fixed change in any parameter value can also be modeled by a switch. For example, in Fig. 6.1(c), S_3 is N.O. and R_3 has a nominal value of 1000Ω . Closing of S_3 means that the value of R_3 is changed from 1000 to 800Ω . Since these are not physical switches, but merely mathematical models representing fault conditions, we shall call them "fault switches" to distinguish from real switches.

6.2 Formulation of Multiport Constraint Equation

With dc stimuli, with all nonlinearities approximated by piecewise linear curves, and with hard failures represented by switches, the networks under consideration can be modeled with linear resistors, controlled sources, dc independent sources, switches, and ideal diodes. Let there be t test nodes whose voltages are to be calculated. It is convenient to view these voltages \mathbf{v}_t as those associated with t zero-valued independent current sources ($i_t \equiv 0$). Let there be d ideal diodes, m fault switches. Extract these elements to form a $(t+m+d)$ -port N as shown in Fig. 6.2, with appropriate notation for the voltages and currents.

Using the technique described in [13], we can obtain a set of $(t+m+d)$ independent constraint equations in $2(t+m+d)$ port variables for the multiport. Before

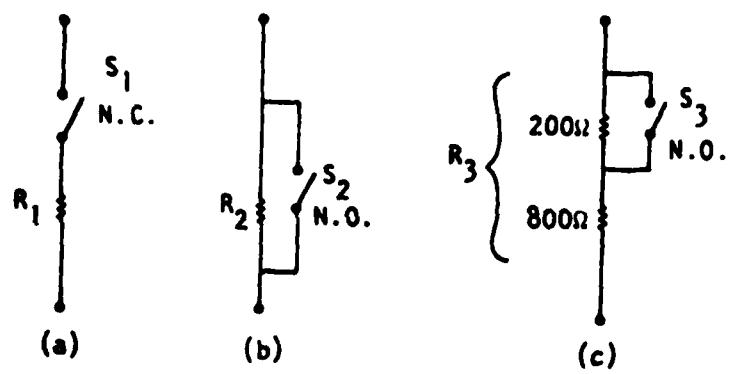


Figure 6.1 Hard failures represented by switches.

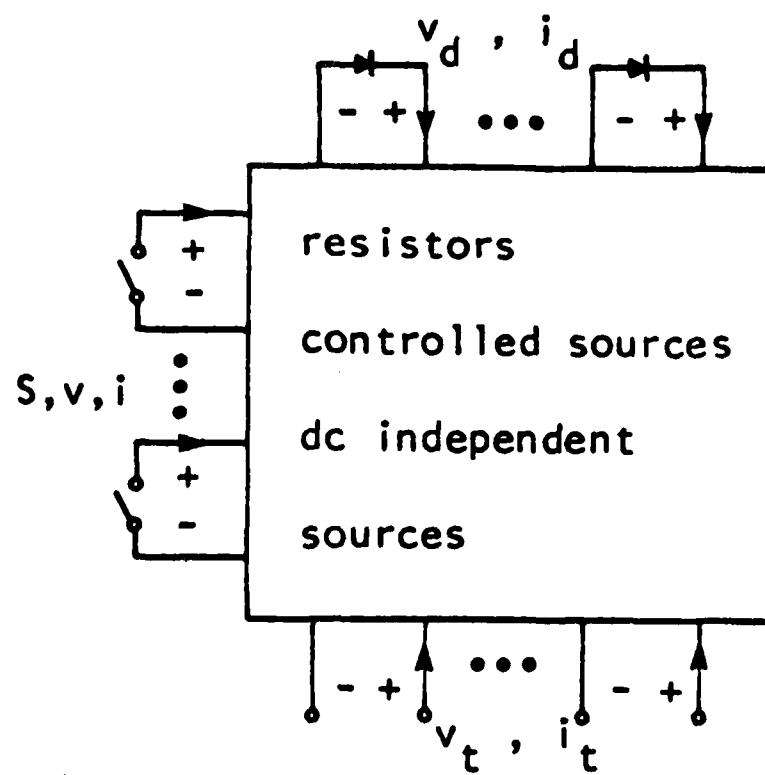


Figure 6.2 Formation of a resistive n-port.

stating this equation, it is necessary to introduce additional notation. For the m fault switches, there are 2^m combinations of switch conditions only one of which represents the nominal circuit while the others represent faults (single as well as multiple faults). For any switch j , either $i_j \equiv 0$ or $v_j \equiv 0$. Therefore, under any particular switch condition, m variables out of (v, i) will be identically zero. We may therefore partition v and i as follows (subscripts NO stands for "normally open" and NC for "normally closed").

$$v = \begin{bmatrix} v_{NO} \\ v_{NC} \end{bmatrix} = \begin{bmatrix} v_{NO} \\ 0 \end{bmatrix}, i = \begin{bmatrix} i_{NO} \\ i_{NC} \end{bmatrix} = \begin{bmatrix} 0 \\ i_{NC} \end{bmatrix},$$

for normal circuit. For the circuit under normal condition, the hybrid equation for the multiport then expresses $(v_{NO}, i_{NC}, v_t, v_d)$ in terms of $(i_{NO}, v_{NC}, i_t, i_d)$ in the following form:

$$\begin{matrix} m \text{ rows} \\ t \text{ rows} \\ d \text{ rows} \end{matrix} \begin{bmatrix} 1 & 0 & 0 & C_{14} & C_{15} & C_{16} \\ 0 & 1 & 0 & C_{24} & C_{25} & C_{26} \\ 0 & 0 & 1 & C_{34} & C_{35} & C_{36} \end{bmatrix}$$

$$x((v_{NO} \ i_{NC}) \ v_t \ i_d (i_{NO} \ v_{NC}) \ i_t \ v_d)^t = [f_1 \ f_2 \ f_3]^t \quad (6.1)$$

where the right hand side constant vector is due to the dc independent sources inside the multiport.

6.3 Calculation of Test Node Voltages

Two different cases of using Eq. (6.1) will now be considered.

(1) *Analysis of the nominal circuit.* Since $i_{NO} \equiv 0$, $v_{NC} \equiv 0$, $i_t \equiv 0$, the corresponding columns in Eq. (6.1) may be discarded. Since the variables (v_{NO}, i_{NC}) are not needed for the fault dictionary, the first row may also be discarded. The equation then simplifies to

$$\begin{bmatrix} 1 & 0 & C_{26} \\ 0 & 1 & C_{36} \end{bmatrix} \begin{bmatrix} v_t \\ i_d \\ v_d \end{bmatrix} = \begin{bmatrix} f_2 \\ f_3 \end{bmatrix} \quad (6.2)$$

From Eq. (6.2), one may write the following two equations

$$v_t = -C_{26}v_d + f_2 \quad (6.3)$$

and

$$i_d = -C_{36}v_d + f_3 \quad (6.4)$$

Eq. (6.4) together with the ideal diode constraints $v_d \geq 0, i_d \geq 0, v_d i_d = 0$ form a complementary problem which may be solved by the use of Lemke's complementary pivot algorithm. After i_d has been found, we calculate the test node voltage vector \mathbf{v}_t by the use of Eq. (6.3).

(2) *Analysis of fault circuits*

Some normally-open switches may now be closed, and some normally-closed switches may now be open. Let \mathbf{x} denote the voltages of the closed switches, and the currents of the open switches of S . Let \mathbf{y} denote the complementary variables of \mathbf{x} (i.e., if x_j is v_j , then y_j is i_j , and vice versa). Then obviously $\mathbf{x} \equiv 0$. Equation (6.1) may be rewritten with \mathbf{x} taking the place of (i_{NO}, v_{NC}) , and \mathbf{y} taking the place of (v_{NO}, i_{NC}) . This of course calls for some *interchanges* between the columns of (i_{NO}, v_{NC}) and the corresponding columns of (v_{NO}, i_{NC}) . Such operations, however, require only a routine bookkeeping. No additional information other than Eq. (6.1) is needed. The new equation will be of the following form;

$$\begin{bmatrix} D_{11} & 0 & 0 & D_{14} & C_{15} & C_{16} \\ D_{21} & 1 & 0 & D_{24} & C_{25} & C_{26} \\ D_{31} & 0 & 1 & D_{34} & C_{35} & C_{36} \end{bmatrix}$$

$$\mathbf{x}[\mathbf{y} \ v_t \ i_d \ \mathbf{x} \ i_t \ v_d]^t = [f_1 \ f_2 \ f_3]^t \quad (6.5)$$

Since $\mathbf{x} \equiv 0, i_t \equiv 0$, the corresponding columns in Eq. (6.5) may be discarded. This leads to the following simplified equation:

$$\begin{bmatrix} D_{11} & 0 & 0 & C_{16} \\ D_{21} & 1 & 0 & C_{26} \\ D_{31} & 0 & 1 & C_{36} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ v_t \\ i_d \\ v_d \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (6.6)$$

Except for pathological circuits, the square submatrix D_{11} is nonsingular. Pivoting on D_{11} , we obtain from Eq. (6.6):

$$v_t = (-C_{26} + D_{21}D_{11}^{-1}C_{16})v_d + (f_2 - D_{21}D_{11}^{-1}f_1) \quad (6.7)$$

$$i_d = (-C_{36} + D_{31}D_{11}^{-1}C_{16})v_d + (f_3 - D_{31}D_{11}^{-1}f_1) \quad (6.8)$$

Eq. (6.8) together with the ideal diode constraints $v_d \geq 0, i_d \geq 0, v_d i_d = 0$ form a complementary problem which may be solved by Lemke's complementary pivot algorithm. After i_d has been found, v_t is calculated from Eq. (6.7).

Remarks:

(1) The data base required for the compilation of the fault dictionary is the following matrix of $(m+d+t) \times (m+d+1)$.

$$\begin{array}{c}
 \begin{array}{l}
 m \text{ rows} \\
 t \text{ rows} \\
 d \text{ rows}
 \end{array}
 \left[\begin{array}{ccc}
 C_{14} & C_{16} & f_1 \\
 C_{24} & C_{26} & f_2 \\
 C_{34} & C_{36} & f_3
 \end{array} \right] \\
 \begin{array}{l}
 m \text{ col.} \\
 d \text{ col.} \\
 1 \text{ col.}
 \end{array}
 \end{array} \quad (6.9)$$

A very comprehensive list of hard failures may be considered initially to set up the data base given by (6.9). Later on, if only a subset of the faults are deemed likely, then only a portion of (6.9) needs to be considered.

(2) In actual computer implementation, Eqs. (6.7) and (6.8) will be derived from Eq. (6.6) by row echelon reduction, instead of finding D_{11}^{-1} explicitly.

(3) When the fault circuits are arranged in a sequence such that two successive fault circuits differ by just one switch condition, then the solution of the second circuit can be obtained quite easily from the first one. The total time spent in pretest analysis is therefore considerably less than that required by the use of a circuit simulator.

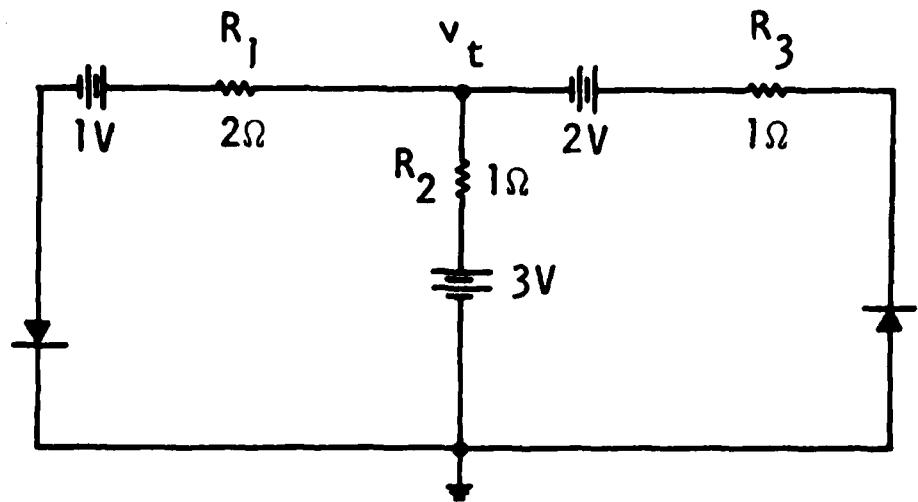
6.4 A Simple Example

A numerical example will now be given to illustrate the procedures and notation. The circuit is deliberately chosen to be simple so that all steps can be verified without the aid of a digital computer.

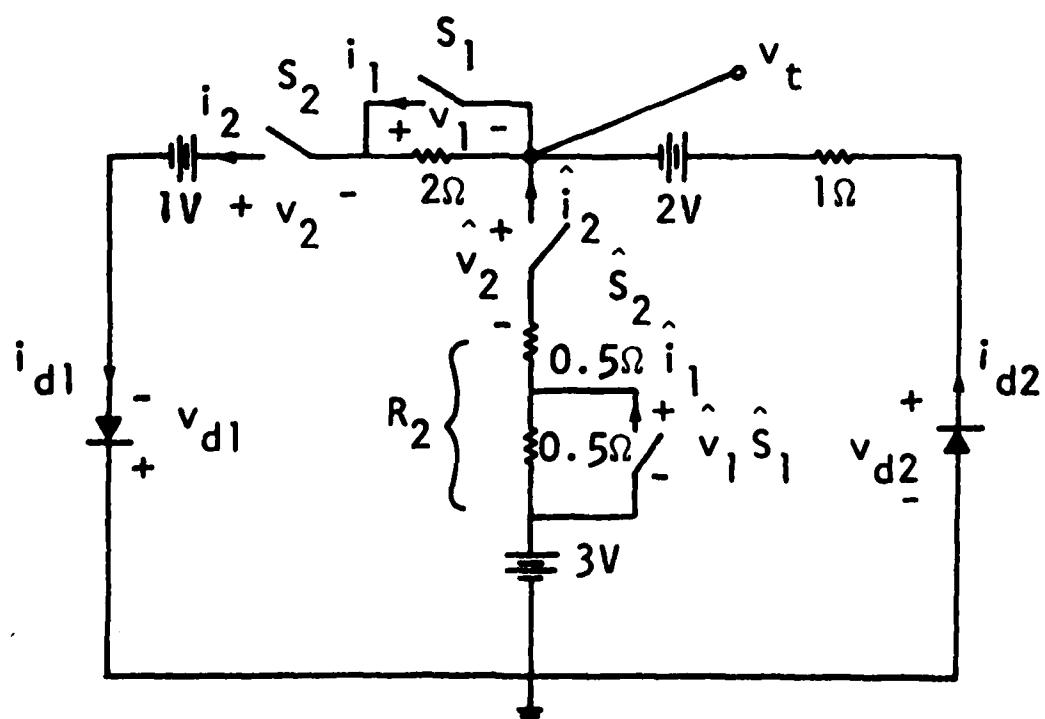
Figure 6.3(a) shows a circuit in the normal operating condition. Suppose that the faults consist of R_1 shorted (F_1), R_1 open (F_2), R_2 open (F_3), and R_2 changing from 1Ω to 0.5Ω (F_4).

The first step is to model these faults with switches S_1, S_2, \hat{S}_1 and \hat{S}_2 as shown in Fig. 6.3(b). Under the normal condition, we have S_1, \hat{S}_1 open, and S_2, \hat{S}_2 closed. It is desired to determine the test node voltage v_t corresponding to each of these faults.

The 4 fault switches, 2 ideal diodes, and 1 test node-pair are extracted to form a 7-port. A straightforward circuit analysis leads to the following equation corresponding Eq. (6.1). (Note that $m = 4$, $t = 1$, and $d = 2$. Voltage and current notations are shown in Fig. 6.3(b)).



(a)



(b)

Figure 6.3 A simple circuit example.

$$l_7 \begin{bmatrix} -0.35 & 0.3 & 0.2 & 0.1 & -0.2 & 0.1 & -0.2 \\ 0.3 & 0.6 & 0.4 & 0.2 & 1.0 & 0.2 & -0.4 \\ -0.2 & -0.4 & 0.4 & -0.8 & -0.4 & -0.8 & -0.4 \\ -0.1 & -0.2 & -0.8 & -0.4 & -0.2 & -0.4 & -0.2 \\ -0.2 & -0.4 & 0.4 & 0.2 & -0.4 & 0.2 & -0.4 \\ -0.1 & -0.2 & -0.8 & -0.4 & -0.2 & -0.4 & -0.2 \\ 0.2 & 0.4 & -0.4 & -0.2 & 0.4 & -0.2 & -0.6 \end{bmatrix}$$

$$x[\dot{v}_1 \dot{i}_2 v_1 i_2 v_t i_{d1} i_{d2} \dot{i}_1 \dot{v}_2 i_1 v_2 i_t v_{d1} v_{d2}]^t \quad (6.10)$$

$$= [-1.2, -2.4, -0.4, -0.2, 0.6, -0.2, -2.6]^t$$

For the fault F_1 , we have (S_1, S_2, \hat{S}_2) closed, and \hat{S}_1 open. The equation corresponding to (6.6) is

$$\begin{bmatrix} 0.4 & 0 & 0 & 0 & 0 & -0.8 & -0.4 \\ -0.8 & 1 & 0 & 0 & 0 & -0.4 & -0.2 \\ 0.4 & 0 & 1 & 0 & 0 & 0.2 & -0.4 \\ -0.8 & 0 & 0 & 1 & 0 & -0.4 & -0.2 \\ -0.4 & 0 & 0 & 0 & 1 & -0.2 & -0.7 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ v_t \\ i_{d1} \\ i_{d2} \\ v_{d1} \\ v_{d2} \end{bmatrix} = \begin{bmatrix} -0.4 \\ -0.2 \\ 0.6 \\ -0.2 \\ -2.6 \end{bmatrix} \quad (6.11)$$

Observe that first column in (6.11) is obtained from column 10 for i_1 in Eq. (6.10). Elementary row operations are now applied to Eq. (6.11) to reduce it to Eq. (6.12).

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -0.2 & -0.1 \\ 0 & 1 & 0 & 0 & 0 & -2 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -2 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ v_t \\ i_{d1} \\ i_{d2} \\ v_{d1} \\ v_{d2} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \\ -3 \end{bmatrix} \quad (6.12)$$

From (6.12), we obtain the following equation corresponding to Eq. (6.8)

$$\begin{bmatrix} i_{d1} \\ i_{d2} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} v_{d1} \\ v_{d2} \end{bmatrix} + \begin{bmatrix} -1 \\ -3 \end{bmatrix} \quad (6.13)$$

This equation together with $i_d \geq 0$, $v_d \geq 0$, $v_d^t i_d = 0$ form the complementary problem. Solution by Lemke's complementary pivot algorithm gives $i_{d1} = 2$, $v_{d1} = 0$, $i_{d2} = 0$, and $v_{d2} = 3$.

The test node voltage v_t , given by Eq. (6.7), is obtained from Eq. (6.12)

$$v_t = v_{d1} + 1 = -0 + 1 = 1$$

The values of v_t for other faults are determined in like manner.

7. FAULT ISOLATION AND REDUCTION OF TEST NODES

After we have calculated the dc voltages of all test nodes, under all fault conditions and input stimuli, our next task is to determine whether each fault can be uniquely identified or not, given a set of measured test-node voltages. This process is usually called *fault isolation*. In Section 2.2, we described the concept of ambiguity set, and two rules (as used in [4]) to obtain maximum isolation and to reduce the number of test nodes. We shall continue to use the ambiguity set concept as defined in [4]. But our method for manipulating the ambiguity sets is quite different from and more systematic than the two rules used in [4]. Instead of describing our method in general terms, one shall illustrate it with a specific (although hypothetical) example.

7.1 Construction of the Table of Ambiguity Sets

Suppose that for a given circuit the preselected fault list consists of 8 faults designated F1 through F8. The nominal circuit condition is designated F0 or NOM. One input vector and 4 test nodes (V1, V2, V3, V4) have been chosen. The test node voltages have been determined by the method of Sections 6 or by the use of a circuit simulation program. The origin of the data is immaterial for the intended fault isolation. The voltages are tabulated below.

	F0	F1	F2	F3	F4	F5	F6	F7	F8
V1	5.0	7.0	7.4	7.3	7.2	9.6	9.7	9.8	5.2
V2	9.0	5.0	6.0	6.4	6.2	5.1	5.2	5.3	9.2
V3	9.5	6.0	6.1	6.2	8.0	6.3	6.4	5.3	4.0
V4	5.0	5.1	5.2	8.8	6.0	6.1	9.0	5.3	6.2

Table 7.1 Calculated voltages of all test nodes.

For each test node we can define its ambiguity sets voltage ranges and circuit conditions. This process becomes very clear if we present the information in Table 7.1 in the form of plots as shown in Fig. 7.1. Circuit conditions that belong to the same ambiguity set correspond to the points in Fig. 7.1 that form a "cluster". With the aid of Fig. 7.1, we can easily construct the following table for ambiguity sets.

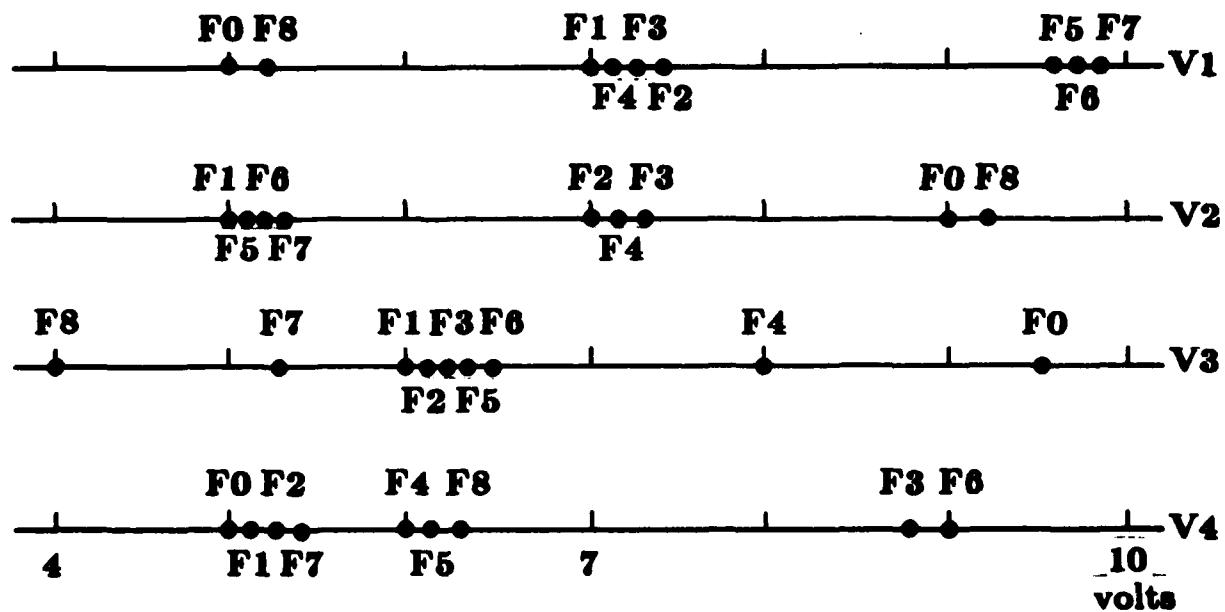


Figure 7.1 Formation of ambiguity sets.

(Node, amb. set)	circuit conditions	voltage range
(1,1)	F0, F8	4.4 -- 5.9
(1,2)	F1, F2, F3, F4	6.4 -- 7.8
(1,3)	F5, F6, F7	9.0 -- 10.4
(2,1)	F1, F5, F6, F7	4.5 -- 5.6
(2,2)	F2, F3, F4	5.7 -- 6.9
(2,3)	F0, F8	8.4 -- 9.8
(3,1)	F8	3.3 -- 4.4
(3,2)	F7	4.6 -- 5.7
(3,3)	F1, F2, F3, F5, F6	5.8 -- 6.9
(3,4)	F4	7.3 -- 8.7
(3,5)	F0	8.7 -- 10.2
(4,1)	F0, F1, F2, F7	4.5 -- 5.5
(4,2)	F4, F5, F8	5.7 -- 6.8
(4,3)	F3, F6	8.2 -- 9.6

Table 7.2 Ambiguity sets contents and voltage ranges.

Note that node 3 has 5 ambiguity sets, while nodes 1, 2, 4 each has 3 ambiguity sets.

The range of each ambiguity set in Table 7.2 is determined in the following manner. First, the center of each cluster is taken to be the average of the two extreme values of the cluster. Next a range of ± 0.7 volt from the center is *tentatively* set. After the *tentative* ranges for all ambiguity sets have been calculated, a check is made to see whether the ranges of any two ambiguity sets (of the same test node) overlap. If so, both ranges are reduced by an equal amount, until a gap or 0.1 or 0.2 volt is obtained. After such revisions, the ranges are accepted for use. As an example, consider ambiguity sets (3,2) and (3,3) in Table 7.2. The second ambiguity set of node 3 has only one value of 5.3 volts (corresponding to F7), which is then also the center value. Therefore, ambiguity set (3,2) has a *tentative* range from 4.6 to 6.0 volts. The third ambiguity set of node 3 has two extreme values 6.0 and 6.4 (corresponding F1 and F6, respectively), and hence a center value of 6.2 volts. Therefore ambiguity set (3,3) has a *tentative range from 5.5 to 6.9*. These two ranges overlap. We decrease the upper boundary of set (3,2) from 6.0 to 5.7, and increase the lower boundary of set (3,3) from 5.5 to 5.8. Each range is reduced by 0.3 volt, and a gap of 0.1 volt has been created. This explains how the range from 5.8 - 6.9 is obtained for ambiguity set (3,3). Similar adjustments are made for all ambiguity set ranges.

7.2 Fault Isolation by Intersection of Ambiguity Sets

Suppose that one ambiguity set of some test node VJ contains only one circuit condition FK, then whether that circuit condition has occurred can be determined by measuring node voltage VJ only. In this case, we say that fault FK has been isolated. For example, let the measured value of V3 for the above hypothetical circuit be 8.2 volts. Then Table 7.2 indicates that the value belongs to ambiguity set (3,4). Since this set has only one element, namely F4, the circuit must be under the condition of fault #4.

On the other hand, if we measure one node voltage only, and obtain $V_3 = 6.5$ volts we will not be able to isolate the fault. Table 7.2 indicates that 6.5 volts belongs to ambiguity set (3,3), and the circuit may be under any one of the following conditions: F1, F2, F3, F5, or F6. Clearly, to isolate the fault, more test nodes must be used.

Let us see how the use of additional test nodes can help isolating the faults. We shall disregard Rule 1, but carry out Rule 2 of [4] (see also Section 2.2) more systematically by constructing an "ambiguity sets intersection table" as shown in Table 7.3 below. To avoid verbosity, we simply say "intersection of VJ and VK" when we really mean the intersection of ambiguity sets of these nodes.

V4 \ V3	F0,F1 F2,F7	F4,F5 F8	F3,F6
F8	ϕ	F8	ϕ
F7	F7	ϕ	ϕ
F1,F2,F3 F5,F6	F1,F2	F5	F3,F6
F4	ϕ	F4	ϕ
F0	F0	ϕ	ϕ

Table 7.3 Intersection of ambiguity sets of V3 and V4.

In the above table the row headings contain ambiguity sets of nodes 3, whereas the column headings contain those of node 4. Each (i,j) block of the matrix contains the result of intersection of ambiguity sets (3,i) and (4,j), with ϕ denoting a null set. For example, the content of the (3,1) block is determined as follows:

$$\begin{aligned} & \text{amb. set (3,3)} \cap \text{amb. set (4,1)} \\ &= \{F1,F2,F3,F5,F6\} \cap \{F0,F1,F2,F7\} = \{F1,F2\} \end{aligned}$$

We note that this intersection yields a total of 7 ambiguity sets: {F8}, {F7}, {F4},

$\{F_0\}$, $\{F_5\}$, $\{F_1, F_2\}$, $\{F_3, F_6\}$. In particular, F_5 has been isolated with the addition of test node 4. For example, if the measured values are $V_3 = 6.5$ volts and $V_4 = 6.2$ volts, then the circuit condition belongs to ambiguity sets (3,3) and (4,2), according to Table 7.2. But Table 7.3 shows that the only fault which occurs in both ambiguity set (3,3) and (4,2) is F_5 . Therefore we conclude that the circuit is under condition F_5 .

With two test nodes V_3 and V_4 , we have isolated faults F_4 , F_5 , F_7 , F_8 - a 50% isolation. We note that F_1 has not been separated from F_2 , neither has F_3 from F_6 . More test nodes must be used to achieve a higher percentage of isolation.

Suppose that we decide to add test node V_1 with the hope of resolving the ambiguity in $\{F_1, F_2\}$, and also in $\{F_3, F_6\}$. The effect of adding V_1 can be seen from the following intersection table. An ambiguity set with only one element is called a *singleton*. The intersection of a singleton with another ambiguity set is trivially simple - either a null set or the singleton itself. To save space, operations involving singletons are not shown in detail in the intersection tables (except in Table 7.3).

$\backslash V_1$	F_0, F_8	F_1, F_2	F_5, F_6, F_7
(V_3, V_4)		F_3, F_4	
F_1, F_2	ϕ	F_1, F_2	ϕ
F_3, F_6	ϕ	F_3	F_6
5 singletons		5 singletons	

Table 7.4 Intersection of (V_3, V_4) with V_1

Observe that F_3 and F_6 have been isolated, but F_1 and F_2 have not. Let us further add test node 2. The intersection operation is shown in Table 7.5 below.

$\backslash V_2$	F_1, F_5	F_2, F_3, F_4	F_0, F_8
(V_3, V_4, V_1)	F_6, F_7		
F_1, F_2	F_1	F_2	ϕ
7 singletons		7 singletons	

Table 7.5 Intersection of (V_3, V_4, V_1) with V_2 .

The faults F_1 and F_2 are now isolated. This indicates that if we use all of the 4 test nodes, we can have 100% fault isolation. But the result does not imply that one *must* use all 4 test nodes to achieve 100% isolation. For example, if the node to be added after (V_3, V_4) is V_2 , instead of V_1 , we will have the following intersection table. instead of V_1 , we will have the following intersection table.

		V2	F1,F5	F2,F3,F4	F0,F8
			F6,F7		
(V3,V4)	F1,F2	F1	F2	ϕ	
	F3,F6	F6	F3	ϕ	
5 singletons		5 singletons			

Table 7.6 Intersection of (V3,V4) with V2.

The result indicates that 100% fault isolation can also be achieved without test node 1.

7.3 Reduction of the Number of Test Nodes

In the general case, the determination of a *minimum* set of test nodes to achieve the highest percentage (not always 100%) of isolation is a very time-consuming process for large circuits. It can be shown that the computation time is not polynomial-bound. Fortunately, for practical applications, we need not insist on getting the *theoretical minimum* number of test nodes. Any *near minimum* solution will serve our purpose, if the solution is simple. In other words, a heuristic method might be more useful for solving practical problems.

We propose to use the following heuristic procedures for reducing the number of test nodes.

Procedure 1.

Step 1. Select the node that has the largest number of ambiguity sets. If a tie occurs, arbitrarily select one among them.

Step 2. Select the next node whose intersection with previously selected nodes will result in the *largest number* of ambiguity sets. In case of a tie, arbitrarily select one.

Step 3. If the number of the resultant ambiguity sets is equal to the number of circuit conditions (no. of faults + 1, the one being for the nominal case), stop. Otherwise go to Step 2.

Consider the previous example of Table 7.2. Node 3 is selected first, because it has 5 ambiguity sets, the largest among V1, V2, V3 and V4. Next, one node from V2, V3, and V4 is to be selected. The intersection of V3 with V4 yield a total of 7 ambiguity sets (see Table 7.3). Similarly $V_3 \cap V_1$ yields 6 sets, and so does $V_3 \cap V_2$. Therefore V4 is chosen as the second test node. The third test node is to be selected from V1 and V2. Now $(V_3 \cap V_4) \cap V_2$ yields 9 ambiguity sets (see Table 7.6) whereas $(V_3 \cap V_4) \cap V_1$ yields only 8 sets (see Table 7.4). Therefore V2 is chosen. With V3, V4, V2 chosen as the test nodes, 100% fault isolation is achieved. Test node V1 is seen

to be redundant.

Procedure 1 will in most cases lead to a *near optimum* selection of test nodes (in fact, an optimum selection in the previous example). But even this procedure may be too time-consuming for large circuits. This is because that in Step 2, every node has to be intersected with the previously selected group of nodes. Therefore, a further simplified procedure is proposed below.

Procedure 2.

Step 1. Select the node that has the largest number of ambiguity sets. If a tie occurs, arbitrarily select one among them.

Step 2. In the remaining nodes, *tentatively* select one having the largest number of ambiguity sets. If a tie occurs, pick any one among them. Now obtain the intersection of this node, VJ, with the previous selected group of nodes. If the intersection increases the total number of ambiguity sets, then select VJ as a test node. Otherwise, disregard VJ.

Step 3. If the number of resultant ambiguity sets is equal to the number of circuit conditions, stop. Otherwise, go to Step 2.

Let us illustrate Procedure 2 with the previous example of Table 7.2. In Step 1, we select V3 since it has 5 ambiguity sets, the largest among V1, V2, V3, V4. Each of the remaining nodes, V1, V2, V4 has 3 ambiguity sets. According to Step 2, we may arbitrarily pick one. Suppose that node 1 is picked. The intersection $V3 \cap V1$ has 6 ambiguity sets, one more than that of V3 alone. Therefore, V1 is selected as the second test node. The third test node is to be arbitrarily selected from V2 and V4. Suppose that we tentatively pick V2. The intersection $V2 \cap (V3 \cap V1)$ has 7 ambiguity sets, one more than that of $(V3 \cap V4)$. Therefore V1 is selected as the third test node. There are a total of 9 circuit conditions (1 nominal plus 8 faulty conditions). Since $7 < 9$, we go through Step 2 another time and include V4 as the fourth test nodes. As shown previously, with V1, V2, V3 and V4 all selected as test nodes, we achieve 100% fault isolation. But this clearly is not an optimum solution, since 3 test nodes V3, V4, V2 will achieve the same goal.

In most cases Procedure 2 will produce a near optimum solution to the selection of test nodes. Since its computational effort is much less than that of Procedure 1, we have implemented Procedure 2 in our present digital computer program.

In reference [4], a distinction is made between "fault detection" and "fault isolation". In our present approach, this distinction requires no additional computational effort. We include the nominal circuit (designated by F0 or NOM) in the ambiguity set manipulations. Separation of a fault FJ from NOM amounts to fault detection, while separation among faults is the usual fault isolation.

What happens if all test nodes have been used, and still less than 100% fault isolation is achieved? In this case, the first question to be investigated is whether further fault isolation is necessary. If several unseparated faults belong to the same SRU (shop replaceable unit), then further isolation is really unnecessary, since the whole unit has to be replaced anyway.

Suppose that further fault isolation is necessary. Then there are two avenues open to solve the problem.

(1) Increase the number of test nodes. Usually, an engineer's familiarity and experience with the circuit may help in the selection of additional test nodes. In the absence of such experience, we may initially choose all *accessible nodes* as the test nodes, and use Procedure 1 or Procedure 2 described above to eliminate unnecessary nodes.

(2) Use more input vectors, i.e., test the circuit under more input conditions. Very little work has been reported in the literature about the design of stimuli for fault isolation. This is a topic for which much research needs to be done.

Assume that additional input vectors have been decided upon. Then the ambiguity set manipulations will have to be modified slightly. Our strategy (as implemented in the digital computer program) is to consider first each node separately. The ambiguity sets of each node under different input vectors are "intersected" to obtain the "all-input ambiguity sets". After such "all-input ambiguity sets" have been obtained for all nodes, they are processed according to Procedure 1 and Procedure 2 to select the desired test nodes. The details of the multiple input-vector case is discussed in a separate technical report [18].

8. AN INTEGER-CODE FAULT DICTIONARY AND POST-TEST FAULT IDENTIFICATION

8.1 *Compilation of the Fault Dictionary*

Once the test nodes have been selected using Procedure 1 or Procedure 2 described in Section 7, we can determine a unique integer code for each circuit condition. For this purpose we need only the ambiguity sets table as given by Table 7.2. Each fault FJ belongs to exactly one ambiguity set of every test node. This information can be tabulated as shown in Table 8.1 below for the hypothetical example of Table 7.2. We assume that V3, V4, V2 have been selected as the test nodes.

Circuit condition	integer code		
	V3	V4	V2
F0	5	1	3
F1	3	1	1
F2	3	1	2
F3	3	3	2
F4	4	2	2
F5	3	2	1
F6	3	3	1
F7	2	1	1
F8	1	2	3

Table 8.1 Integer codes for all faults

As an illustration, consider the code 312 for F2. This means that F2 is in the third ambiguity set of V3, the first set of V4, and the second set of V2. Other codes are interpreted in the same manner.

As pointed out in Sec. 7.1, fault F4 can be identified by measuring voltage V3 alone, since F4 is the only element in ambiguity set (3,4). Thus, we could have given F4 an integer code of the form 4XX, where X means "don't care". The code 4XX for F4 implies that measurements of V4 and V2 are unnecessary (don't care) for the identification of F4. For the same reason, the following alternative codes could have been assigned to all faults (other possibilities exist):

Circuit condition	integer code		
	V3	V4	V2
F0	5	X	X
F1	3	1	1
F2	3	1	2
F3	3	3	2
F4	4	X	X
F5	3	2	X
F6	3	3	1
F7	2	X	X
F8	1	X	X

Table 8.2 Integer fault codes including "don't care"

At first glance, it looks as though the use of Table 8.2 is more preferable to Table 8.1, because some measurements (don't care) need not be done. But in our work we have preferred Table 8.1 for two reasons:

- (1) Once the test connections are available, taking additional voltage readings involves very little additional effort.
- (2) The use of Table 8.1 provides more certainty about fault location than the use of Table 8.2.

The integer codes in Table 8.1 may now be arranged in the dictionary order (ascending numbers) as shown in Table 8.3 below, which is what we call an integer-code fault dictionary.

Fault code			Faults
V3	V4	V2	
1	2	3	F8
2	1	1	F7
3	1	1	F1
3	1	2	F2
3	2	1	F5
3	3	1	F6
3	3	2	F3
4	2	2	F4
5	1	3	F0

Table 8.3 A fault dictionary using test nodes V3, V4, and V2.

Since the rightmost column has only one fault for each code, 100% fault isolation has been achieved. Had we used only two test nodes V3 and V4, the following fault dictionary would have resulted

Fault code		Faults
V3	V4	
1	2	F8
2	1	F7
3	1	F1,F2
3	2	F5
3	3	F3,F6
4	2	F4
5	1	F0

Table 8.4 A fault dictionary using test nodes V3 and V4

The above fault dictionary shows that F1 and F2 are not isolated, nor are F3 and F6. As mentioned earlier further isolation may not be necessary if the unseparated faults are located within the same "shop replaceable unit".

The fault dictionary set-up for the multiple input vectors case is discussed in [18].

8.2 Post-test Fault Identification by Dictionary Look Up

In Section 2 we reviewed the fault dictionary approach of [4]. The fault dictionary consists of the calculated voltages of all test nodes under all fault conditions. After the test on a faulty circuit has been made, the post-test analysis to identify the fault involves the calculation of many "sum of squared deviations". See Section 2.3.

The most attractive feature of our proposed integer-code fault dictionary is that no arithmetic operations are needed after test. We only have look up the dictionary to identify the fault (or a group of faults). Since the dictionary entries are arranged in a prescribed order (ascending integers in the present case), the looking-up operation becomes very simple. We will now illustrate the procedure with a specific example.

Let us continue with the hypothetical example of Table 7.2. Suppose that a test on the faulty circuits yields the following readings:

$$V_3 = 6.2, V_4 = 8.5, V_2 = 6.0 \quad (\text{volts})$$

Referring to Table 7.2, we see that the reading of V_3 falls in ambiguity set 3, V_4 in set 3, and V_2 in set 2. Therefore, the circuit condition has an integer code 332. Now refer to the fault dictionary, Table 8.3. There is only one fault associated with the code 332. Therefore we conclude that fault F_3 has occurred.

What happens if the voltage reading of some test node does not fall in any of its ambiguity set ranges? If we use the fault dictionary approach of [4], there is a danger of being led to a wrong conclusion. This is because that one of the faults, although not the actual fault, may produce a minimum SSD (see Section 2.3) small enough for us to believe that it has occurred. In contrast, using our present approach, we will not draw any conclusion about the circuit condition - a case of "no decision". Such a situation indicates that either the fault list (of hard failures) is not comprehensive enough, or the fault is not a hard failure. This former case can be remedied by expanding the fault list and recompile the fault dictionary. The latter case of soft failures requires a completely different approach for diagnosis. Volume 2 of this technical report describes a new ac multifrequency diagnosis method for soft failure [19].

In the fault dictionary described above, the code for each fault is of the form

$$I_1 I_2 \cdots I_N$$

where N is equal to the number of test nodes, and the largest value for each integer I_j is equal to the number of ambiguity sets of its associated test node. The decision rules used to identify the fault are in the form of multiple level logic. This is quite convenient if the fault dictionary look-up operation is *manual*, i.e., a person actually turns the page and finds the codes. But this operation can also be made completely *automatic*, i.e., the technician need only feed in the test data, a decision making circuitry will process the data and display the fault number F_j automatically. To achieve

this, the multilevel logic must be converted into binary logic. The details of this conversion and a scheme for hardware implementation is described in [18].

9. A COMPLETE VIDEO AMPLIFIER CIRCUIT EXAMPLE

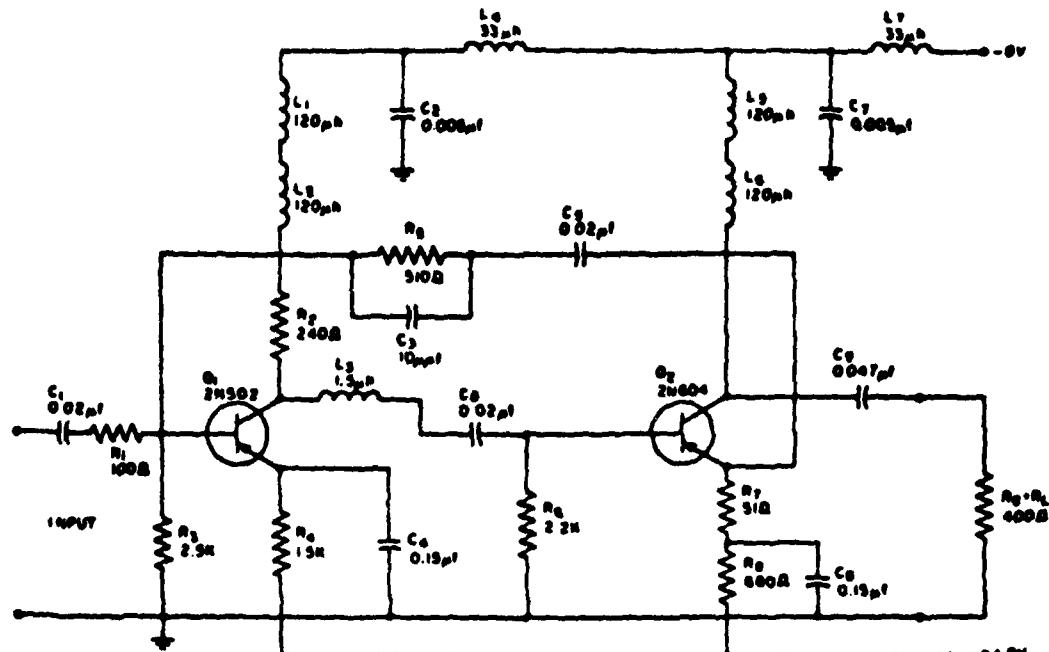
With the sole purpose of explaining the basic principles of our new approaches, we have in Section 4 to 8 used very simple examples so that each example can be worked out in full. Our ultimate aim of course is to implement our techniques and apply to realistic circuits. We have written a digital computer program that employs full matrix technique and handles medium size networks up to the size of a video amplifier which we use as the test circuit in this initial stage of the project. Since the main concern at this point is demonstrating the feasibility of the approach, no effort has been made to optimize the computer codes, nor has any consideration been given to make the program user orientated. The documentation of this program is given in [20].

We shall now illustrate our new approach to dc fault diagnosis using the computer outputs generated for a practical circuit.

Figure 9.1(a) shows the schematic diagram of a video amplifier. The engineer, based on his experience, preselects 16 faults for detection. The list of faults is given in Table 9.1. Note that the nominal case is just one of the many circuit conditions, and is listed as case #1 in Table 9.1

number	description
1	nominal case
2	L1 and/or L2 open
3	L4 open
4	L3 open
5	L7 open
6	L5 and/or L6 open
7	Q1 base open
8	Q2 base open
9	C4 shorted
10	C6 shorted
11	C3 shorted
12	C5 shorted
13	C8 shorted
14	C9 shorted
15	Q1, B-E shorted
16	Q2, B-E shorted

Table 9.1 Definition of Faults



(a)

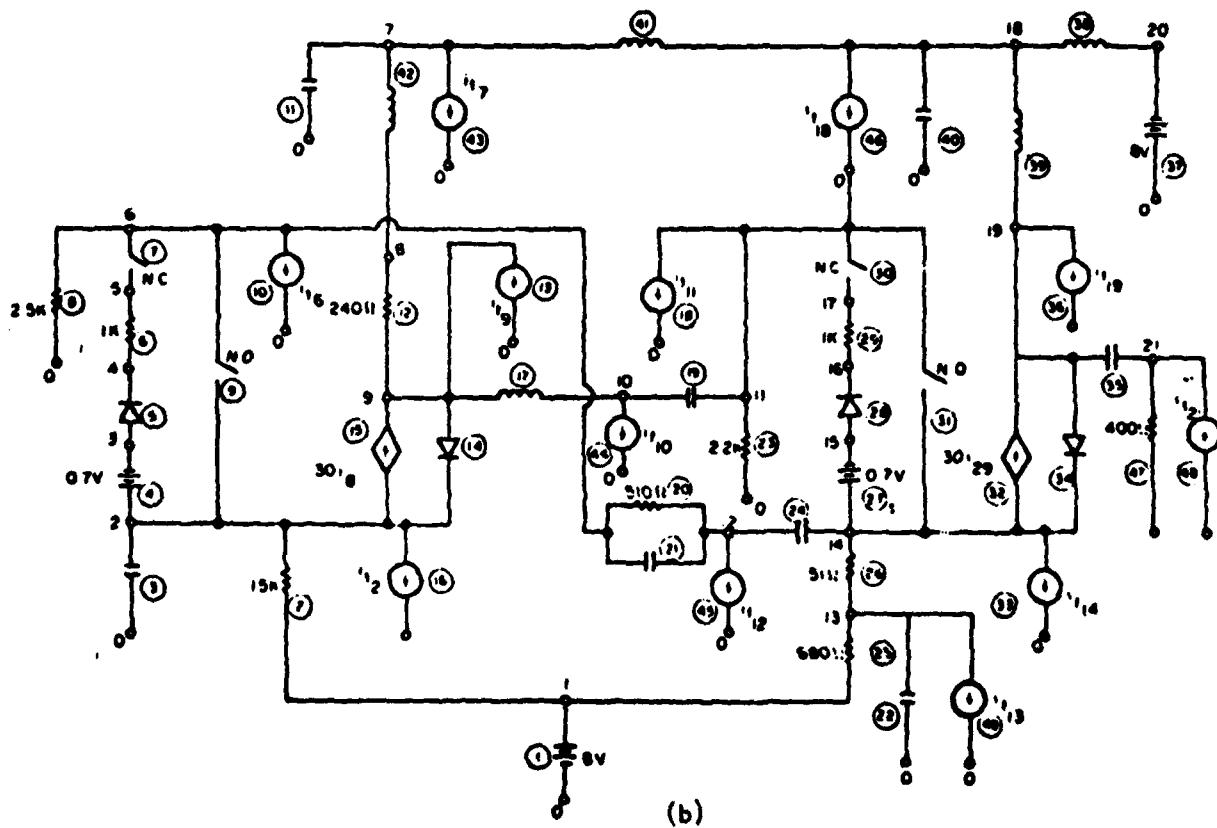


Figure 9.1 A video amplifier and its PWL model for fault analysis.

The piecewise linear models for transistors Q1 and Q2 are next obtained, with the fault conditions Q1BES, Q2BES, Q1BO, Q2BO represented by switches. For analysis purpose, the circuit schematic is replaced by the piecewise linear model shown in Figure 9.1(b). The 11 independent current sources in Figure 9.1(b) actually have zero values. They are required by the n-port theory to facilitate the calculation of the test nodes voltages. Initially, 11 test nodes are chosen and they are the nodes numbered 2, 6, 7, 9, 11, 12, 13, 14, 18, 19 and 21.

The method used to analyze the piecewise linear network is fully explained in Section 6. In Table 9.2, we show a portion of the computer output that contains all test node voltages under all fault conditions.

	F1	F2	F3	F4	...
V2	1.21	5.81	5.18	1.21	
V6	3.65	3.65	3.65	3.65	
V7	-8.00	-8.00	-5.81	-8.00	
V9	-6.95	5.81	5.81	-6.95	
.					
.					
.					

Table 9.2 Node voltages for all faults.

Next, the ambiguity sets at each test node are determined using the algorithm similar to that described in Section 7. The computer output is a table of ambiguity set ranges. The portion for test node 2 is shown in Table 9.3. Similar tables are printed out by the computer for all test nodes.

ambiguity set	voltage	range	fault cases
1	0.51	1.91	1,4,6,8,10,11,13,14,16
2	5.11	6.51	2,3,5
3	7.30	8.00	7
4	-0.70	0.70	9
5	1.48	2.88	12
6	4.30	5.70	15

Table 9.3 Ambiguity sets for node 2.

As shown above, there are 6 ambiguity sets for node 2. Ambiguity set 2 is from 5.11 to 6.51 volts, and contains F2, F3, F5.

The algorithm described in Section 7 is used to process the above tables with the aim of reducing the number of test nodes to near minimum. The computer output from this algorithm shows that in the present case only 5 test nodes voltages are necessary: V2, V11, V13, V7, and V21. The algorithm at the same time generates the integer codes for all faults. These computer generated codes are then sorted in ascending order to form the fault dictionary as shown in Table 9.4.

Fault code (V2, V11, V13, V7, V21)	Fault number
1 1 1 1 1	1,4,11
1 1 1 1 2	14
1 2 2 1 1	6
1 1 4 1 1	16
1 3 1 1 1	8
1 4 3 1 1	10
1 5 1 1 1	13
2 1 1 1 1	2
2 1 1 2 1	3
2 2 2 2 1	5
3 1 1 1 1	7
4 1 1 1 1	9
5 1 1 1 1	12
6 1 1 1 1	15

Table 9.4 Fault dictionary for the video amplifier.

To illustrate the use of this fault dictionary, suppose that we have a faulty video amplifier with the following measured node voltages:

$V_2 = 1.2$, $V_{13} = 8.0$, $V_{11} = 0.5$, $V_7 = -8.0$, $V_{21} = -8.0$ volts.

Referring to Table 9.3, we find that the fault is in ambiguity set 1 for node 2. Similarly, referring to the remaining tables (not shown here, but given in [18], we find the fault is in set 3 for node 13, set 1 for node 11, set 1 for node 7 and set 1 for node 21. Hence the integer code is 13111. Now referring to Table 9.4, we find that this corresponds to fault number 8. Further referring to Table 9.1, we see that the fault has been determined as F8, i.e., transistor Q2 base open.

10. CONCLUSION AND FURTHER RESEARCH

DC fault diagnosis has been found quite useful for identifying hard failures which account for about 80% of failures in analog electronic equipment [4]. In order to enhance of capability of the dc fault dictionary approach we have in this research project attacked the problem with new techniques: These include:

- (1) Model all nonlinearities by piecewise linear characteristics.
- (2) Use switches to represent open circuits and short circuits.
- (3) Solve the piecewise linear resistive network containing switches by the use of multiport theory and Lemke's complementary pivot algorithm.
- (4) Use multilevel logic to reduce the number of test nodes and to generate integer fault codes for the fault dictionary.

A digital computer program [20] has been written to implement these techniques. The program employs full matrix technique and can adequately handle medium size circuits such the video amplifier circuit shown in Section 9.

Although the feasibility of our new approach has been established, much research remains to be done before user oriented software and hardware can be produced to handle larger circuits (at least one order of magnitude larger than the video amplifier circuit of Section 9). We list below several topics we consider as important for future studies.

- (1) Investigate the use of sparse matrix techniques in all phases of the computation in order to handle larger circuits.
- (2) Implement the post-test analysis technique developed in this project in *hardware*, so that after the measured voltages are received, the ATE will display the number of fault case automatically without any human intervention.
- (3) Investigate the construction of piecewise linear macro models for "shop replaceable units."
- (4) Devise more effective fault isolation algorithms. This study naturally includes the selection of test nodes and the design of input stimuli.
- (5) Investigate the nondeterministic aspects of dc fault diagnosis. This study will consider the device parameters as random variables instead of fixed values. At present, the only precaution we have against measurement inaccuracies and device parameter variation is by widening the gaps between adjacent ambiguity sets. Apparently, a more elegant theory and a more powerful computational techniques is needed.

REFERENCES

- [1] N. Navid and A. N. Wilson "A theory and an algorithm for analog circuit fault diagnosis," *IEEE Trans. on Circuits and Systems*, Vol. CAS-26, pp. 440-457, July 1979.
- [2] P. M. Lin, "A survey of symbolic network functions," *IEEE Trans. on Circuits and Systems*, Vol. CT-20, pp. 732-737, Nov. 1973.
- [3] V. Visvanathan and A. Sangiovanni-Vincentelli, "Diagnosability of nonlinear circuits and systems - Part 1: The dc case," *IEEE Trans. on Circuits and System*, Vol. CAS-28, pp. 1093-1102, November 1981.
- [4] W. Hochwald and J. D. Bastian, "A dc approach for analogue fault dictionary determination," *ibid*, pp. 523-529.
- [5] A. Pahwa and R. A. Rohrer, "Band faults: Efficient approximation to fault bands for the simulation before fault diagnosis of linear circuits," *IEEE Trans. on Circuits and Systems*, Vol. CAS-29, pp. 81-88, May 1982.
- [6] N. Sen and R. Saeke, "Fault diagnosis for linear systems via multifrequency measurements," *IEEE Trans. on Circuits and Systems*, Vol. CAS-26, pp. 457-465, July 1979.
- [7] R. E. Tucker and L. P. McNamee, "Computer-aided design application to fault detection and isolation techniques," *Proc. 1977 IEEE Int'l Symp. on Circuits and Systems*, pp. 684-687, April, 1977.
- [8] T. N. Trick, W. Mayeda, and A. A. Sakla, "Calculation of parameter values from node voltage measurements," *IEEE Trans. on Circuits and Systems*, Vol CAS-26, pp. 466-474, July 1979.
- [9] L. W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Memo ERL-M520, Electronic Research Laboratory, University of California, Berkeley, May 1975.
- [10] "ASTAP General Information Manual," GH20-1271-0, IBM Corp., 1973.
- [11] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.
- [12] L. O. Chua, *Introduction to Nonlinear Network Theory*. New York: McGraw-Hill, 1969.
- [13] L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, N.J.: Prentice-Hall, 1975.

- [14] D. T. Phillips, A. Ravindran, and J. J. Solberg, *Operations Research: Principles and Practice*. New York: Wiley, 1976.
- [15] I. Getreu, *Modeling the Bipolar Transistor*. Tektronix, Inc., 1978.
- [16] H. J. Zimmermann and S. J. Mason, *Electronic Circuit Theory: Devices, Models, and Circuits*. New York: Wiley, 1959.
- [17] S. N. Stevens and P. M. Lin, "Analysis of piecewise linear resistive networks using complementary pivot theory," *IEEE Trans. on Circuits and Systems*, Vol. CAS-28, pp. 429-441, May 1981.
- [18] Y. Elcherif and P. M. Lin, "An isolation algorithm for analog fault dictionary," School of Electrical Engineering, Purdue University, technical report (to appear).
- [19] L. Rapisarda and R. DeCarlo, "Fault diagnosis of nonlinear analog circuits: Vol. 2, A multifrequency method for soft failure analysis" Sch. of Elec. Engrg., Purdue University, Rep. TR-EE-82-22, July 1982.
- [20] Y. Elcherif and P. M. Lin, "A computer program for dc fault diagnosis," Sch. of Elec. Engrg., Purdue University, technical report (to appear).

